
THE UNIVERSITY
*of* EDINBURGH

# Target Localization Using RSS Measurements in Wireless Sensor Networks

Zeyuan Li


**Thesis submitted for the degree of**

**Doctor of Philosophy**

**The University of Edinburgh**

**School of Engineering**

**Year of Submission 2016**

# Abstract

The subject of this thesis is the development of localization algorithms for target localization in wireless sensor networks using received signal strength (RSS) measurements or Quantized RSS (QRSS) measurements.

In chapter 3 of the thesis, target localization using RSS measurements is investigated. Many existing works on RSS localization assumes that the shadowing components are uncorrelated. However, here, shadowing is assumed to be spatially correlated. It can be shown that localization accuracy can be improved with the consideration of correlation between pairs of RSS measurements. By linearizing the corresponding Maximum Likelihood (ML) objective function, a weighted least squares (WLS) algorithm is formulated to obtain the target location. An iterative technique based on Newtons method is utilized to give a solution. Numerical simulations show that the proposed algorithms achieves better performance than existing algorithms with reasonable complexity.

In chapter 4, target localization with an unknown path loss model parameter is investigated. Most published work estimates location and these parameters jointly using iterative methods with a good initialization of path loss exponent (PLE). To avoid finding an initialization, a global optimization algorithm, particle swarm optimization (PSO) is employed to optimize the ML objective function. By combining PSO with a consensus algorithm, the centralized estimation problem is extended to a distributed version so that can be implemented in distributed WSN. Although suboptimal, the distributed approach is very suitable for implementation in real sensor networks, as it is scalable, robust against changing of network topology and requires only local communication. Numerical simulations show that the accuracy of centralized PSO can attain the Cramer Rao Lower Bound (CRLB). Also, as expected, there is some degradation in performance of the distributed PSO with respect to the centralized PSO.

In chapter 5, a distributed gradient algorithm for RSS based target localization using only quantized data is proposed. The ML of the Quantized RSS is derived and PSO is used to provide an initial estimate for the gradient algorithm. A practical quantization threshold designer is presented for RSS data. To derive a distributed algorithm using only the quantized signal, the local estimate at each node is also quantized. The RSS measurements and the local estimate at each sensor node are quantized in different ways. By using a quantization elimination scheme, a quantized distributed gradient method is proposed. In the distributed algorithm, the quantization noise in the local estimate is gradually eliminated with each iteration. Simulations show that the performance of the centralized algorithm can reach the CRLB. The proposed distributed algorithm using a small number of bits can achieve the performance of the distributed gradient algorithm using unquantized data.

# Lay Summary

Motivated by recent developments in localization techniques in Wireless Sensor Networks (WSNs), this thesis aims to develop new localization methods based on received signal strength (RSS) measurements or Quantized RSS (QRSS) measurements.

Chapter 3 studies target localization methods under the assumption that RSS measurements are correlated. In most published work, RSS measurements are assumed uncorrelated while in reality they are correlated. By utilizing the correlation, the accuracy of localization could improve. With the statistical relationship between RSS measurements and the distance between a target and a sensor node, mathematical equations are formulated. Then an iterative technique is utilized to give a solution to the equations. Numerical results show that the proposed algorithms achieves better performance than existing algorithms with reasonable complexity.

Chapter 4 investigates target localization with unknown path loss model parameters. Most existing literatures estimate location and these parameters jointly using iterative methods with a good initialization of path loss exponent (PLE). To avoid finding an initial value, a global optimization algorithm, particle swarm optimization (PSO) is employed to optimize the mathematical equations. By combining PSO with a consensus algorithm, the centralized localization problem is extended to a distributed version so that can be implemented in distributed WSNs. This distributed approach is very suitable for implementation in real sensor networks, as it is scalable, robust against changing of network topology and requires only local communication. Numerical results show that the accuracy of centralized PSO can attain the lower bound on the variance. Also, as expected, there is some degradation in performance of the distributed PSO with respect to the centralized PSO.

Chapter 5 studies distributed algorithms for RSS based target localization using only quantized data. The quantized RSS measurements are modeled following the path loss model under the assumption that the shadowing noise is Gaussian. A practical quantization threshold designer is presented for RSS data. To derive a distributed algorithm using only the quantized signal, the local estimate at each sensor node is also quantized. A scheme to eliminate the round-off error in the quantized local estimate is also proposed. Numerical results show that the proposed distributed algorithm using a small number of bits can achieve the performance of the distributed gradient algorithm using unquantized data.

# Declaration of Originality

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the School of Engineering at The University of Edinburgh.

Zeyuan Li

# Acknowledgements

First of all, I would like to thank my supervisors Prof. Bernard Mulgrew and Dr. Pei-Jung Chung for inspiration and technical guidance. Without their help I could never have gone this far. Second, thanks to my parents, whose love and support warm my heart. Third, thanks to my wife for her considerate care for during the past one and half years. Finally, thanks to my friends in Edinburgh who help me in work and everyday life.

# Contents

# List of Figures

# List of Tables

# Acronyms and Abbreviations

| | |
|---|---|
| AOA | Angle of Arrival |
| ADC | Analogue to digital converters |
| ARMSE | Average Root Mean Square Error |
| CDF | Cumulative Density Function |
| CRLB | Cramer Rao Lower Bound |
| CWLS | Constrained Weighted Least Squares |
| DG | Distributed Gradient |
| DPSO | Distributed Particle Swarm Optimization |
| DRSS | Differential Received Signal Strength |
| DV-Hop | Distance Vector Hop |
| DwSDP | Distributed weighted Semidefinite Programming |
| EA | Evolutionary Algorithm |
| EM | Expectation Maximization |
| FIM | Fisher information matrix |
| GA | Genetic Algorithm |
| GPS | Global Positioning system |
| GTRS | Generalized Trust Region Subproblem |
| KF | Kalman Filter |
| LLS | Linear Least Squares |
| LOS | Line of Sight |
| LP | Linear Programming |
| LS | Least Squares |
| MAP | Maximum a Posteriori |
| MDS | Multidimensional Scale |
| ML | Maximum Likelihood |
| NLOS | None Line of Sight |

| | |
|---|---|
| NLS | Nonlinear Least Squares |
| NRMSE | Normalised Root Mean Square Error |
| PDF | Possibility Density Function |
| PLE | Path Loss Exponent |
| PMF | Possibility Mass Function |
| POCS | Projection onto Convex Sets |
| PSO | Particle Swarm Optimization |
| QDG | Quantized Distributed Gradient |
| QRSS | Quantized Received Signal Strength |
| RF | Radio Frequency |
| RMSE | Root Mean Square Error |
| RSS | Received Signal Strength |
| SA | Simulated Annealing |
| SDP | Semidefinite Programming |
| SI | Swarm Intelligence |
| SOCP | Second-order Programming |
| TDOA | Time Difference of Arrival |
| TLS | Total Least Squares |
| TOA | Time of Arrival |
| UKF | unscented Kalman filter |
| UT | Unscented transform |
| WLS | Weighted Least Squares |
| WSN | Wireless Sensor Network |

# Nomenclature

| | |
|---|---|
| $\mathbf{A}$ | Matrix in the LLS of RSS based localization |
| $\mathbf{A}_g$ | Adjacency matrix |
| $b$ | Length of surveillance area |
| $\mathbf{b}$ | A vector with distance information in the LLS of the RSS based localization |
| $\hat{\mathbf{b}}$ | $\mathbf{b}$ with noise |
| $\Delta\hat{\mathbf{b}}$ | The noise in $\mathbf{b}$ |
| $c$ | Penalty coefficient |
| $\mathbf{c}_i$ | Position of sensor node $i$ |
| $\mathbf{C}$ | Error Covariance matrix |
| $d_0$ | Reference distance |
| $d_i$ | Distance between target and node $i$ |
| $\hat{d}_i$ | Unbiased estimate of $d_i$ |
| $\hat{d}_{\text{ML}_i}$ | The ML estimate of $d_i$ |
| $\mathbf{d}$ | Distance vector of $d_i$ |
| $\mathbf{D}_g$ | Degree Matrix |
| $E$ | Edge |
| $\mathbf{E}$ | Set of edge |
| $f_i$ | Local cost function at node $i$ |
| $\mathbf{g}_b$ | Global best of particles |
| $G_c$ | Communication graph of sensor network |
| $\mathbf{G}$ | Matrix in the CWLS of DRSS based localization |
| $\mathbf{h}$ | Vector in the CWLS of DRSS based localization |
| $\mathbf{H}(\cdot)$ | Wireless Channel |
| $\mathbf{J}$ | Fisher information matrix of RSS based localization |
| $\mathbf{J}_D$ | Fisher information matrix of DRSS based localization |
| $J_z$ | Fisher information on received signal strength $z$ |
| $K_i$ | QRSS measurement at node $i$ |
| $\mathbf{K}$ | QRSS measurement vector |
| $L$ | Number of quantization levels |
| $\mathbf{L}_p$ | Laplacian matrix |

| | |
|---|---|
| $\bar{\mathbf{L}}$ | A diagonal matrix with elements $10^{\frac{P_i - P_0}{10\alpha}}$ |
| $\bar{\mathbf{L}}$ | Laplacian matrix |
| $M$ | Number of quantization bits |
| $M_{\text{run}}$ | Number of Monte Carlo trials |
| $n_i$ | Shadowing noise of RSS measurement |
| $n_{d_i}$ | noise in distance estimate $\hat{d}_i$ |
| $n_{v_{ij}}$ | Shadowing noise of DRSS measurement |
| $N$ | Number of sensor nodes |
| $\mathcal{N}$ | Gaussian distribution |
| $\mathcal{N}_i$ | A set of adjacent nodes of node $i$ |
| $\bar{\mathbf{p}}_{i,k}(t)$ | Best position of particle $i$ has until the step $t$ at dimension $k$ |
| $\mathbf{p}$ | Received signal strength vector |
| $P_s$ | Number of Particles |
| $PL(d)$ | Path Loss at distance $d$ |
| $P_0$ | Transmit Power |
| $P_i$ | Received signal strength at node $i$ |
| $\mathbf{p}$ | Vector of received signal strength |
| $q_{ij}$ | DRSS measurement |
| $\mathbf{q}$ | Vector of DRSS measurement |
| $\mathbf{Q}$ | Covariance matrix of shadowing noise |
| $\mathbf{Q}_s$ | Covariance matrix |
| $\mathbf{Q}_v$ | Covariance matrix of DRSS noise |
| $\mathbf{Q}_w$ | Covariance matrix |
| $r_{ij}$ | Distance ratio of $d_i$ and $d_j$ |
| $\hat{r_{ij}}$ | Estimate of $r_{ij}$ |
| $\Delta r_{ij}$ | Noise in $\hat{r_{ij}}$ |
| $s_i$ | Threshold $i$ of QRSS measurement |
| $\mathbf{s}$ | Threshold vector of QRSS measurement |
| $s(x; y; z; t; f)$ | A space-time-frequency signal |
| $s_T(x_T; y_T; z_T; t_T; f_T)$ | A space-time-frequency signal at transmitter |
| $s_R(x_R; y_R; z_R; t_R; f_R)$ | A space-time-frequency signal at receiver |
| $T_{\text{max}}$ | Maximum iteration number in any iteration algorithm |
| $\mathbf{u}$ | Position of target |
| $\mathbf{u}^{[i]}(t)$ | Estimated position of target at node $i$ in $t$th iteration |

| | |
|---|---|
| $\mathbf{u}^{[i],Q}$ | Quantized estimated position of target at $i$th node |
| $\Delta\mathbf{u}$ | Noise in the estimated $\mathbf{u}$ |
| $\bar{\mathbf{U}}$ | Equals $\mathbf{u}\mathbf{u}^T$ |
| $\hat{\mathbf{U}}$ | A matrix combination of the vectors $\hat{\mathbf{u}}$ |
| $\mathbf{v}_{i,k}$ | Velocity vector of particle $i$ at $k$th dimension |
| $V$ | Vertices |
| $V_{\text{max}}$ | Maximum velocity of particles |
| $w$ | Inertial weight of PSO |
| $w_{\text{min}}$ | Minimum value of $w$ |
| $w_{\text{max}}$ | Maximum value of $w$ |
| $\mathbf{W}$ | Weighting matrix for consensus |
| $x$ | x-coordinate of target |
| $x_i$ | x-coordinate of sensor node $i$ |
| $y$ | y-coordinate of target |
| $y_i$ | y-coordinate of sensor node $i$ |
| $z_i$ | $P_0 - 10\alpha \log_{10} d_i$ |
| $\mathbf{z}$ | Vector of $z_i$ |
| $\alpha$ | Path Loss exponent |
| $\beta$ | Step size for gradient method |
| $\boldsymbol{\gamma}_r$ | A vector used in CWLS of RSS based localization |
| $\boldsymbol{\gamma}_d$ | A vector used in CWLS of DRSS based localization |
| $\boldsymbol{\Gamma}_r$ | First order derivative of $\boldsymbol{\gamma}_r$ with respect to target location |
| $\boldsymbol{\Gamma}_d$ | First order derivative of $\boldsymbol{\gamma}_d$ with respect to target location |
| $\Delta$ | Uniform quantization step size |
| $\sigma_i$ | Standard derivation of shadowing noise at node $i$ |
| $\sigma_u$ | Standard derivation of target location |
| $\sigma_\alpha$ | Standard derivative of PLE |
| $\boldsymbol{\Omega}$ | A matrix in the FIM of DRSS based localization |
| $\boldsymbol{\eta}$ | Variable in Newton's method |
| $\rho_{ij}$ | Correlation coefficient between node $i$ and node $j$ |
| $\mu_\alpha$ | Mean of PLE |
| $\boldsymbol{\theta}$ | Vector of interested parameters |
| $\hat{\boldsymbol{\theta}}_{\text{ML}}$ | The ML estimate of $\boldsymbol{\theta}$ |
| $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ | The MAP estimate of $\boldsymbol{\theta}$ |

| | |
|---|---|
| $\boldsymbol{\vartheta}$ | A vector contains the target location in the LLS of RSS based localization |
| $\vartheta_i$ | $i$th element of $\boldsymbol{\vartheta}$ |
| $\hat{\boldsymbol{\vartheta}}$ | Estimate of $\boldsymbol{\vartheta}$ |
| $\boldsymbol{\varkappa}$ | A vector contains the target location in the CWLS of DRSS based localization |
| $\lambda_i$ | Eigenvalue of $\bar{\mathbf{L}}$ |
| $\tau_i(t)$ | Gradient of local cost function $i$ at iteration $t$ |
| $\psi_1$ | Cognitive component in PSO |
| $\psi_2$ | Social component in PSO |
| $\boldsymbol{\Psi}$ | Variable in Newton's method |
| $\bar{\xi}_i$ | An auxiliary variable in definition of QRSS measurement |
| $\xi_i$ | An auxiliary variable in FIM of QRSS based localization |
| $\zeta_i$ | The gradient of the local cost function $f_i$ |
| $p(\cdot)$ | Probability function |
| $\mathbb{E}(\cdot)$ | Expectation operator |
| $(\cdot)^T$ | Transpose of $(\cdot)$ |
| $\parallel (\cdot) \parallel$ | Norm of $(\cdot)$ |
| $\approx$ | Approximately equal to |
| $\log(\cdot)$ | Logarithm operator |
| $\exp(\cdot)$ | Exponential operator |
| $\det(\cdot)$ | Determinant operator |
| $\partial$ | Partial derivative |
| Var$(\cdot)$ | Variance of $(\cdot)$ |
| Tr$(\cdot)$ | Trace operator |

# Chapter 1

# Introduction

## 1.1 Target Localization in Wireless Sensor Networks

Recent technology improvements have enabled the development of low-cost, low-power and multi-functional sensor nodes. Such nodes can be equipped with a variety of sensors such as temperature, humidity, pressure and motion detectors. Consequence, sensor nodes are capable of detecting environmental conditions. These nodes are also equipped with microprocessors for signal processing and radio frequency circuits for wireless communications. The deployment of a large number of sensor nodes in a certain area constitute a Wireless Sensor Network (WSN) which is capable of collecting and retrieving data from the environment. The WSNs are widely used in forest fire monitoring [1], structure health monitoring [2], object tracking [3], ecological habitat monitoring [4], industrial process control [5], smart houses [6], etc. A brief introduction of WSN is provided in Appendix B.

Localization is a key task (often mandatory) in many applications like positioning, tracking and monitoring. ”In environmental applications, such as water quality monitoring, precision agriculture and indoor air quality monitoring, sensing data without knowing the location is meaningless” [7]. The target to be located is a source or a sensor node which is not aware of its own location. The localization task is to determine the location of the target using the interaction between the target and other sensor nodes with known positions. There has been a large body of research on localization for WSNs over past decades. In biological research, cheap sensor nodes can be deployed in a large area to track animals. Tracking animals can record their behaviour and their interaction with environment. Currently, this is done by using Global Positioning System (GPS) based or directional antenna based collars. But these methods only offer a short lifetime due to high energy cost. Using WSNs, the location data can be processed by local nodes near the targets and forwarding to a base station through multihop routing. WSN based localization are also very suitable in applications such as warehousing and logistics. In warehouse and factories,

boxes and parcels can all be tagged with cheap device such as sensor nodes. Sensor nodes not only can monitor storage conditions, i.e., temperature and humidity, but also can report their location when any box or equipment needs to be found. In these applications, traditional positioning techniques, i.e., GPS are not well suited. GPS can provide accurate location information outdoors. The United States government claims 4 meter root mean square horizontal accuracy for civilian GPS. However, GPS will not work precisely in buildings, tunnels, undergrounds, etc. It will also increase the size and cost of sensor nodes. As estimated in a recent report - "iphone 7: analysis & breakdown", the GPS module costs about 8 dollars. For cheap sensor nodes around 10-20 dollars, 8 dollars are relatively high.

Currently, most localization methods in WSNs can be classified into two categories: range-free methods and the range-based methods. In range-free methods, algorithms do not need the absolute information of the distance or angle between the target and sensor nodes. Range-free algorithms can indirectly obtain the distances between the unknown target and sensor nodes by connectivity information or exchanged multi-hop routing information. Further, such information can be employed to calculate the coordinates of unknown nodes.

Distance-Vector Hop (DV-Hop) [8] is a typical representation of a range-free positioning system. The localization process of DV-Hop has three steps. In the first step, each node counts its hop to the unknown sensor node. Secondly, nodes approximate their distance between themselves and the target node by using the hop counts and average distance in each hop. Thirdly, the system calculate the unknown position using a multilateration method [9]. Another typical range-free method is the centroid method [10]. In this methods, nodes estimate their location as the centroid of their neighbouring nodes' coordinates. The accuracy of the method is about one third the separation among neighbouring nodes, which leads to highly dense networks for practical accuracy. In [11], a two-step procedure is proposed that achieves higher accuracy. In the first step, a coarse estimate of the target's position is obtained using a similar procedure as in [10]. A refinement step then follows considering only those nodes that belong to the one hop neighbourhood of the target. Another category of range-free methods is the region overlap algorithm [12]. This method builds a bounding box around the node where the target should lie and then it obtains the intersection among regions of all nodes. An iterative multilateration refinement process is then employed to obtain the final estimate.

The main advantage of range-free based localization is their low cost in hardware and simplicity in computation. These merits allow them to be easily implemented in WSNs. However, their low

positioning accuracy makes their applications limited, depending on scenario requirements.

In range-based positioning systems, relatively accurate distance information is employed to locate the target. When distance estimates are available, multilateration or Least Squares (LS) can be used to estimate the position. To obtain a distance estimate, different measurements have been proposed in the literature [13], i.e., Time of Arrival (TOA), Time Difference of Arrival (TDOA), Angle of Arrival (AOA) and Received Signal Strength (RSS). TOA measurements can be affected by multipath signals. In line of sight (LOS) scenarios, multipath signal can arrive very soon after the LOS signal causing interference. In non line of sight (NLOS) scenarios, TOA measurements are heavily biased. They may be even incapable of locating the target. Moreover, TOA needs to be synchronized. This will cause extra cost in WSNs. As for RSS measurements, since they are average energy measured during a short period, there is no need for synchronization between target and sensor nodes. Also, RSS will not be easily affected by NLOS. In the literature, RSS can be used to assist TOA measurements to locate the target in NLOS scenarios [14]. AOA measurements provide direction information instead of distance information. It provides localization information complementary to the TOA and RSS measurements [15]. Generally, in high accuracy requirements applications, TOA and AOA are preferred. But they are relatively expensive in device, in particular, AOA needs additional antenna. When low cost takes precedence in priority over accuracy, positioning using RSS are popular since it is relatively inexpensive and simple to implement in hardware, and no additional antenna is required. And the accuracy of RSS based localization can be improved by deploying more cheap nodes. However, employing RSS measurements in localization presents several challenges. The first one is that the RSS measurements and the distance between nodes and target follow a logarithmic relationship, which makes it difficult to linearize or relax the maximum likelihood (ML) cost function. Thus, finding the location is challenging. Several algorithms can be found in the literature to cope with this problem, i.e., LS [16], [17], Semidefinite Programming (SDP) [18], [19]. There are also works in [20], [21] focusing on tracking using RSS measurements. Another challenge is that the channel parameters, i.e., transmit power and Path Loss Exponent (PLE), required in the RSS signal model are difficult to obtain in some scenarios. The transmit power of a target depends on its battery which may vary with time, and the path loss exponent depends on the environment. When the transmit power is unknown, researchers tackle this problem by jointly estimating transmit power and locations using LS [22] and SDP [23]. An alternative method is to employ the Differential RSS (DRSS) [24] measurements in which the transmit power is eliminated by subtracting one measurement from another. When it comes to the unknown path loss exponent, researchers tackle

this problem using iterative methods based on the fact that the path loss exponent is within a certain range of values [25], [26].

An important feature of RSS measurement is that the correlation between each measurement. It has been found in real scenario experiments that the correlation coefficient is 0.2 to 0.5 for outdoor environments [27] and 0.4 to 0.8 for indoor environments [28]. As studied in [29], with consideration of the correlation between the RSS measurements, the localization accuracy increases. However, the majority of the existing works can not utilize the correlation directly. In this thesis, one purpose is to develop low-complexity algorithm that can make use of the correlation between RSS measurements.

It is well known that sensor nodes in WSNs are characterized by limited resources, such as energy and communication bandwidth. As per the IEEE 802.15.4 standard, there can be as many as 65566 nodes in the network and the raw data rate can be 20, 40 or250kb/s per node. If nodes generate data constantly, then bandwidth is limited. One way to save energy is to avoid long range wireless transmission. Distributed processing that requires only local communications and processing helps to reduce the transmission energy. In [30], the RSS-based location estimation problem is relaxed into a semidefinite programming (SDP) problem, and further solved by a consensus based distributed SDP method [31]. The disadvantage of the distributed SDP method [30] is that during the semidefinite relaxation, a large error is introduced into the cost function. Thus the error leads to bad performance. It is clear that distributed algorithm for RSS based localization with higher localization accuracy needs to be developed.

Another way to save energy is to limit the data transmitted in the network. It is desirable that only multibit quantized data is transmitted within the network. However, the majority of existing works assume analog data are available for localization. Motivated by this, the Quantized RSS (QRSS) model and its corresponding Cramer Rao Lower Bound (CRLB) is proposed in [32]. In the QRSS model, measurements are quantized before sending to the centre processing unite of WSN. In the centre unit, the QRSS measurements are used to calculate the target position.

This thesis aims to estimate target location in WSN using RSS measurement and QRSS measurement. Distributed algorithms for RSS-based and QRSS-based localization are also proposed in the context of distributed WSN.

## 1.2 Thesis Structure and Contribution

The thesis investigates the target localization problem using RSS measurements in WSNs.

Chapter 2 introduces basic knowledge to help the understanding of this thesis and background information and this work's research context. The covered background topics will include basic concepts of the RSS signal model, DRSS signal model, QRSS signal model and the theory of consensus-based distributed algorithm. Existing methods used in RSS-based localization are reviewed and categorized.

Chapter 3 to Chapter 5 are the main technical chapters, where the main contribution will be described.

In Chapter 3, target localization using RSS measurement in correlated shadowing is investigated. Most previous work on RSS localization assumes that the shadowing components are uncorrelated. However, in most scenarios, the shadowing components are correlated due to physical obstructions. To solve the RSS localization problem with correlated shadowing, first, we extend the unbiased distance estimate based on uncorrelated noise to a correlated noise. Then, with the unbiased distance estimate, a constrainted weighted least squares (CWLS) estimator is formulated. Further, an iterative technique based on Newton's method is employed to give a solution. The CWLS method is also extended to be used in DRSS model in which the transmit power is unknown. The bias and covariance of the proposed method are derived using perturbation analysis. The localization accuracy of CWLS is shown to be close to the CRLB.

In Chapter 4, RSS-based target localization with unknown PLE is investigated. To avoid finding an initialization or using any prior information, particle swarm optimization (PSO) is employed to optimize the joint ML objective function. By combining PSO with a consensus algorithm, the same estimation problem in a distributed sensor network is also addressed, where each sensor node has access only to the data from its neighbours. Although suboptimal, the proposed approach is very suitable for implementation in real sensor networks, as it is scalable, robust against network topology changes and requires only local communication. The performance and convergence speed of the proposed algorithm is studied through simulations.

In Chapter 5, a distributed consensus-based gradient algorithm for RSS based target localization using only quantized data is proposed. The ML estimator of QRSS is derived and PSO is used to solve the cost function. Moreover, a practical quantization threshold designer is presented for

RSS data. The threshold design method only requires information on the channel characteristic of RSS signal model and the length of the surveillance area. To derive a distributed algorithm using only the quantized signal, the local location estimate at each node is quantized using a uniform quantization threshold. By combining a quantization elimination scheme, a quantized distributed gradient (QDG) method is proposed. In the QDG method, the quantization noise in the local location estimate is gradually eliminated along with every iteration. Simulation shows that the performance of the centralized PSO algorithm can reach the CRLB with a relatively small number of bits. The proposed distributed algorithm using a small number of bits can achieve the performance of the distributed gradient algorithm using unquantized data.

Chapter 6 concludes the thesis and gives suggestions for further investigation.

The work of this thesis is based on the following publications:

- Zeyuan Li and P. Chung, "Weighted Least Squares Solution for RSS based Localization in Correlated Shadowing," In *Signal Processing Systems (SiPS), 2014 IEEE Workshop on*, Belfast, 2014, pp. 80-84.
- Zeyuan Li, Pei-Jung Chung, Bernard Mulgrew, Distributed target localization using quantized received signal strength, In Signal Processing, Volume 134, 2017, Pages 214-223.

# Chapter 2

# Background and Literature Survey

The main advantage for received signal strength (RSS) based localization techniques is its low cost and the fact that most receivers are capable of estimating RSS. The relation between distance and Path Loss depends on the channel behaviour; thus, an accurate propagation model is required for reliable estimations. Based on the well-known log-normal propagation model, the RSS-based localization problem has been widely studid using different estimation and optimization techniques. However, the estimates of the parameters of the RSS model requires a large amount of training data in the calibration process. In scenarios which the system needs to be frequently re-calibrated, techniques are developed to estimate the target location with the parameters of RSS model inaccurate or even unknown. Moreover, to save communication costs, the quantized RSS (QRSS) is proposed in [32]. In the literature, the location estimation problem using QRSS measurements have not been well studied. This chapter provides a background of the thesis and a survey of some existing localization techniques. In Section 2.1, the RSS signal model and existing location estimation algorithms will be reviewed. In Section 2.2, localization algorithms with inaccurate or unknown parameters of log-normal propagation model are reviewed. Then the QRSS signal model will be introduced in Section 2.3. Section 2.4 concludes this chapter.

## 2.1 RSS Signal Model

Wireless communication has evolved significantly, over the past decades, to meet the fast growing demand for high data rates over the wireless medium. The wireless channel plays an key role in the design of wireless communication systems since communication path can be obstructed by objects like trees, buildings, mountains.

The wireless signal propagates in space, based on the laws of physics. An Radio Frequency (RF) signal which travels in a medium suffers an attenuation (path loss) based on the nature of the medium. In the travel, the signal is obstructed by objects results in the signal getting absorbed,

signal traversing multiple paths, signals frequency being shifted. It is clear that the RF signal is a space-time-frequency signal and can be represented as $s(x; y; z; t; f)$ where $x, y, z$ are the space variables and $t, f$ are time and frequency variables, respectively. Denote the transmitted signal as $s_T(x_T; y_T; z_T; t_T; f_T)$ and the received signal as $s_R(x_R; y_R; z_R; t_R; f_R)$, one can relate both space-time-frequency signals as $s_R(x_R; y_R; z_R; t_R; f_R) = \mathbf{H}(s_T(x_T; y_T; z_T; t_T; f_T))$, where $\mathbf{H}(\cdot)$ is a function which can be called the Wireless Channel [33]. The wireless channel is also dependent on factors like atmospheric conditions, mobility of transmitters and receivers, types of antennas used, and other parameters.

The mechanism behind wave propagation are diverse, but can generally be attributed to reflection, diffraction and scattering [34]. Reflection occurs when a propagating wave impinges upon an object which has very large dimensions when compared to the wavelength of the propagating wave. Diffraction occurs when the radio path is obstructed by object which has sharp edges. This may cause the waves to bend around the object. When the medium in which the wave travels is consists of many small objects with dimension that are small compared to the wavelength, scattering occurs. Due to multiple reflections in the communication path, the waves travel along different paths of different lengths. The small scale fading is caused by signal transmission from the transmitter to the receiver via multiple paths. Therefore, it causes constructive and destructive interference. Such interference varies in the spatial scale of several carrier wavelength and hence is called small scale fading [35]. As a object moves very small distances, the received signal strength at this instance many fluctuate rapidly causing small scale fading. Due to reflection, diffraction, scattering, the received signal strength may come from different directions. So the phases are random, thus the sum of signal varies significantly. In small scale fading, small movement may results in large variation in signal magnitude. The large scale fading, on the other hand, is caused by large objects in the signal paths, blocking and reflecting the radio waves and as a result causing a reduction of signal strength [35]. As an object moves a larger distance away form transmitter, the average received signal strength decreases. With these factors, the wireless channels are random and difficult to model. A channel can be modelled physically by trying to calculate the physical processes which modify the transmitted signal. A communication channel can also be modelled statistically. The radio propagation model generally focus on the relationship between received signal strength and the distance between the transmitter and receiver. So they can be used to predict the mean signal strength for an arbitrary transmitter-receiver distance which is useful in estimating the radio coverage area of a transmitter. In this thesis, we wish to locate an object based on the received signal strength which will not fluctuate rapidly, thus the large scale fading model

is employed. Over time, some classical large scale propagation models have emerged, which are now used to predict large scale coverage for communication systems design. Both theoretical and measurement-based models implies that average received signal power decreases algorithmically with distances [34]. The average path loss between transmitter and receiver is expressed as a function of distance by using a path loss exponent (PLE), $\alpha$.

$$PL(d) = P_0 - 10\alpha \log\Big(\frac{d}{d_0}\Big) \tag{2.1}$$

where $d$ is the distance between the transmitter and the receiver, $PL(d)$ (in dBm) is the received signal strength at receiver, $d_0$ is the reference distance which is determined from measurements close to the transmitter and $P_0$ is the average received signal strength at distance $d_0$ from the transmitter. The PLE $\alpha$ represents signal attenuation as a positive quantity measured in dB. In most literature, $P_0$ is referred as the transmit power. It is important to select the reference distance $d_0$ that is best for the propagation environment. $d_0 = 1$m is commonly used in the existing literature [36]. For notation simplicity, we assume $d_0 = 1$ in remaining parts of the thesis.

However, the signal model (2.1) does not consider the fact that environmental clutter may be different at two different positions having the same distance between transmitter and receiver. Thus, the measured signal strength value is different from the value calculated using (2.1). Measurements have shown that the received signal strength at a particular location is random and distributed normally (in dB) about the value predicted in (2.1) [37], [38]. That is

$$PL(d) = P_0 - 10\alpha \log\Big(\frac{d}{d_0}\Big) + n \tag{2.2}$$

where $n$ represents the random shadowing effects of the wireless channel occurring at locations with same distance but different level of clutter in their propagation path. As introduced in the beginning of this section, the source of noise are diverse, due to the Central limit theorem, the noise on path loss model is more likely to be Gaussian. Data measured in in-door and outdoor environment [34] shows that the random variable $n$ can be approximated by a zero-mean Gaussian random variable with following distribution

$$f(n) = \frac{1}{\sqrt{2\pi\sigma}} \exp\Big(-\frac{n^2}{\sigma^2}\Big), \tag{2.3}$$

| Environment | Path Loss Exponent, $\alpha$ |
|---|---|
| Free space | 2 |
| Urban area cellular radio | 2.7 to 3.5 |
| Shadowed urban cellular radio | 3 to 5 |
| In building line-of-sight | 1.6 to 1.8 |
| Obstructed in building | 4 to 6 |
| Obstructed in factories | 2 to 3 |

Table 2.1: Path loss exponent for different environments.

where $\sigma$ is the standard deviation of $n$.

The estimation of channel parameter $P_0$, $\alpha$ and $\sigma$ can be straight forward. The value of $P_0$ can be measured directly at 1m away from the target. The accuracy of $\alpha$ dominate the accuracy of the path loss model, so a large amount of RSS measurements over a wide range of measurement locations and distance between transmitters and receivers are required. Then, the value of $\sigma$ and $\alpha$ is computed using linear regression. Estimates of $\sigma$ and $\alpha$ can be obtained using a Maximum Likelihood (ML) estimator. Some typical values of PLE are listed in Table 2.1 [34].

Assume a Wireless Sensor Network (WSN) consists of $N$ nodes with known position $\mathbf{c}_i = [x_i, y_i]^T, i = 1, \ldots, N$ and one target with unknown locations $\mathbf{u} = [x, y]^T$. The centre unit of the WSN is aware of each node's position. Assume that the target emits an radio frequency signal that can be heard by all nodes in the network. Following (2.2), the RSS measurement $P_i$ (in dBm) at node $i$ from the target can be expressed as

$$P_i = P_0(d_0) - 10\alpha \log_{10}\left(\frac{d_i}{d_0}\right) + n_i, \quad i = 1, \ldots, N, \tag{2.4}$$

where

$$\begin{aligned} d_i &= \| \mathbf{u} - \mathbf{c}_i \| \\ &= \sqrt{(x - x_i)^2 + (y - y_i)^2} \end{aligned} \tag{2.5}$$

$d_i$ is the distance between the target and node $i$. In the following subsections, some existing methods for location estimation using RSS measurements will be reviewed.

### 2.1.1 Maximum Likelihood Estimator

ML is the most popular estimation approach due to its applicability in complicated estimation problems. The basic principle of ML is simple: the ML estimate for the parameters to be estimated is the value that maximizes $p(\mathbf{p};\mathbf{u})$, where $\mathbf{p}$ is the received signal strength vector $\mathbf{p} = [P_1, \cdots, P_N]$. Assuming the measurements are uncorrelated, the shadowing noise vector $[n_1, \ldots, n_N]^T$ is jointly Gaussian with zero mean and covariance

$$\mathbf{Q} = \text{diag}\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_N^2\}. \tag{2.6}$$

Therefore, the received signal strength vector $\mathbf{p}$ is also Gaussian distributed, shown as:

$$\mathbf{p} \sim \mathcal{N}(\mathbf{z}, \mathbf{Q}). \tag{2.7}$$

where

$$\mathbf{z} = [z_1, \cdots, z_N]^T, \tag{2.8}$$

with element

$$z_i = P_0 - 10\alpha \log_{10} d_i, \tag{2.9}$$

The probability of the given RSS measurements vector $\mathbf{p}$ can be obtained as

$$p(\mathbf{p};\mathbf{u}) = \frac{1}{(2\pi)^{N/2}\sqrt{\det(\mathbf{Q})}} \exp[-\frac{1}{2}(\mathbf{p}-\mathbf{z}(\mathbf{u}))^T\mathbf{Q}^{-1}(\mathbf{p}-\mathbf{z}(\mathbf{u}))]. \tag{2.10}$$

Accordingly, the log-likelihood function can be written as

$$f(\mathbf{p};\mathbf{u}) = \ln\Big(\frac{1}{(2\pi)^{N/2}\sqrt{\det(\mathbf{Q})}}\Big) - \frac{1}{2}(\mathbf{p}-\mathbf{z}(\mathbf{u}))^T\mathbf{Q}^{-1}(\mathbf{p}-\mathbf{z}(\mathbf{u})). \tag{2.11}$$

Notice that the first term in (2.11) does not depend on $\mathbf{u}$. Therefore, maximising (2.11) is equivalent to minimising

$$\hat{\mathbf{u}}_{\mathrm{ML}} = \arg\min_{\mathbf{u}} (\mathbf{p} - \mathbf{z}(\mathbf{u}))^T \mathbf{Q}^{-1} (\mathbf{p} - \mathbf{z}(\mathbf{u})). \tag{2.12}$$

The solution of the ML estimator (2.12) can be found by the numerical search method such as Gauss-Newton [39], [40] or steepest descent [41] methods. The iterative algorithms require a good initialization to guarantee that the algorithm converges to the global minimum. Also, it cannot easily be linearized or relaxed into convex problems. In the Section 2.1.3, we will review some existing techniques that recruit Least Squares (LS) and semidefinite programming (SDP).

### 2.1.2 Cramor Rao Bound

After an estimation of the parameter $\hat{\mathbf{u}}$ is obtained, it is important to know the error covariance matrix. The error covariance matrix is defined as

$$\mathbf{C} = \mathbb{E}[(\hat{\mathbf{u}} - \mathbf{u})(\hat{\mathbf{u}} - \mathbf{u})^T]. \tag{2.13}$$

where $\mathbb{E}[\cdot]$ represents the expectation operator. The Cramor Rao Lower Bound (CRLB) establishes a lower bound on the error covariance matrix for any unbiased estimator of the parameter $\mathbf{u}$. For an unbiased estimator $\hat{\mathbf{u}}$ of the 2-dimensional parameter $\mathbf{u}$, the error covariance matrix for $\hat{\mathbf{u}}$ is bounded as follows:

$$\mathbf{C} \geq \mathbf{J}^{-1}, \tag{2.14}$$

provided $\mathbf{J}$ the Fisher information matrix (FIM). The $i, j$ element of the FIM is expressed as

$$\mathbf{J}_{ij} = \mathbb{E}\Big[ - \frac{\partial^2}{\partial \mathbf{u}_i \partial \mathbf{u}_j} p(\mathbf{p}; \mathbf{u}) \Big]. \tag{2.15}$$

For a 2-dimension localization problem $\mathbf{J}$ is

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} \\ \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{bmatrix}. \tag{2.16}$$

Let $\hat{\mathbf{u}}$ be the unbiased estimator of the coordinates of the target $\mathbf{u}$. We define the location variance of the estimator to be $\sigma_u^2$, then the CRLB asserts that

$$\sigma_u^2 \geq [\mathbf{J}^{-1}]_{11} + [\mathbf{J}^{-1}]_{22} \tag{2.17}$$

$$= \frac{\mathbf{J}_{11} + \mathbf{J}_{22}}{\mathbf{J}_{11}\mathbf{J}_{22} - \mathbf{J}_{12}^2}, \tag{2.18}$$

where $[\cdot]_{ij}$ represents the $i, j$ entry of the matrix.

The FIM of the RSS-based localization, derived in [7], is

$$\mathbf{J} = \frac{100\alpha^2}{(\ln 10)^2}\mathbf{\Gamma}_r^T\mathbf{Q}^{-1}\mathbf{\Gamma}_r, \tag{2.19}$$

where

$$\mathbf{\Gamma}_r = \begin{bmatrix} \frac{x-x_1}{d_1^2} & \frac{y-y_1}{d_1^2} \\ \vdots & \vdots \\ \frac{x-x_N}{d_N^2} & \frac{y-y_N}{d_N^2} \end{bmatrix}. \tag{2.20}$$

### 2.1.3 Localization Approaches

#### 2.1.3.1 Distance Estimate

Recall that RSS based localization is actually a range-based localization problem. As $P_i$ is a function of $d_i$, it is possible to estimate the distance $d_i$ using $P_i$. When $d_i$ is available, the RSS based localization problem is similar to the time of arrival (TOA) based localization problem. Thus, the RSS based localization problem could be solved using TOA based localization algorithms. Fig.2.1 illustrates how the range-based localization problem is solved.

Recall that the measurement noise follows a Gaussian distribution. Therefore, it is clear that the received power at the $i$th node follows a Gaussian distribution with distance dependent mean as

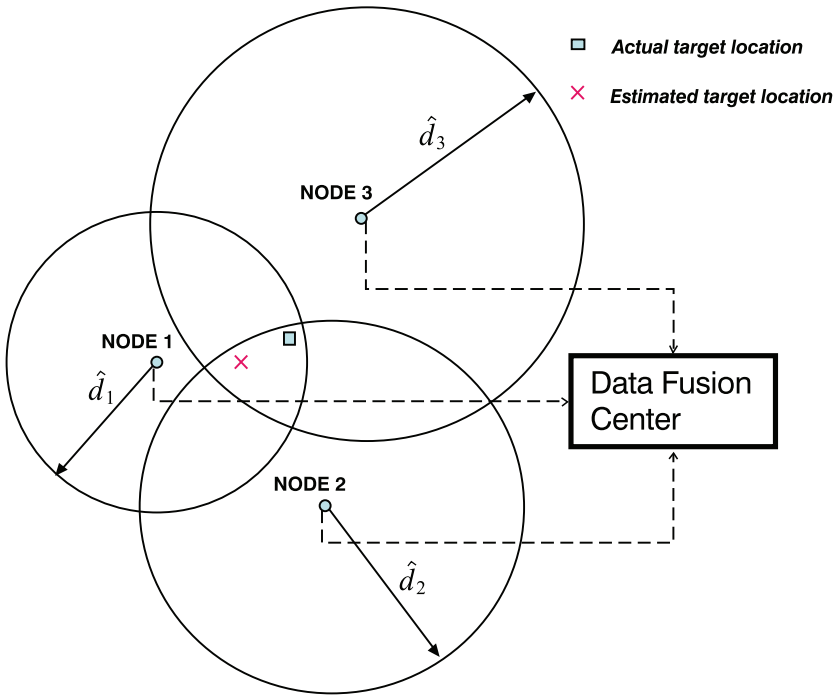



Figure 2.1: Illustration of a simple scenario of distance based localization.

$$p(P_i; d_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\Big( -\frac{1}{2}\Big(\frac{P_i - P_0 + 10\alpha \log_{10} d_i}{\sigma_i}\Big)^2\Big). \tag{2.21}$$

Taking the logarithm of (2.21) and differentiating it with respect to $d_i$ gives

$$\begin{aligned} \frac{\partial \log p(p_i; d_i)}{\partial d_i} &= -\Big(\frac{P_i - P_0 + 10\alpha \log_{10} d_i}{\sigma_i}\Big) \frac{\partial (P_0 - 10\alpha \log_{10} d_i)}{\partial d_i} \\ &= \Big(\frac{P_i - P_0 + 10\alpha \log_{10} d_i}{\sigma_i}\Big)\Big(10\alpha \frac{1}{d_i \ln 10}\Big). \end{aligned} \tag{2.22}$$

where $\partial$ represents the partial derivative. Equating (2.22) to zero requires that

$$P_i = P_0 - 10\alpha \log_{10} d_i. \tag{2.23}$$

Therefore, the ML estimate of the distance between the target and node $i$ is given by

$$\hat{d}_{\mathrm{ML}_i} = 10^{\frac{P_0 - P_i}{10\alpha}}. \tag{2.24}$$

Then, the expectation of $\hat{d}_{\mathrm{ML}_i}$ is

$$\begin{aligned}\mathbb{E}[\hat{d}_{\mathrm{ML}_i}(P_i)] &= \int \hat{d}_{\mathrm{ML}_i} p(P_i; d_i)\,\mathrm{d}P_i \\ &= \int 10^{\frac{P_0-P_i}{10\alpha}} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\Big(-\frac{1}{2}\big(\frac{P_i - z_i}{\sigma_i}\big)^2\Big)\,\mathrm{d}P_i.\end{aligned} \tag{2.25}$$

Expanding the square in (2.25) gives

$$\begin{aligned}\mathbb{E}[\hat{d}_{\mathrm{ML}_i}] &= 10^{\frac{P_0}{10\alpha}} \int \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\Big(-\frac{P_i^2 + z_i^2 - 2z_iP_i + 2\frac{\sigma_i^2 \ln 10}{10\alpha}P_i}{2\sigma_i^2}\Big)\mathrm{d}P_i \\ &= 10^{\frac{P_0}{10\alpha}} \exp\Big(\frac{1}{2}\Big(\sigma_i\frac{\ln 10}{10\alpha}\Big)^2 - \frac{\ln 10}{10\alpha}z_i\Big) \int \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\Big(-\frac{1}{2}\Big(\frac{P_i - z_i + \sigma_i^2\frac{\ln 10}{10\alpha}}{\sigma_i}\Big)^2\Big)\mathrm{d}P_i \\ &= 10^{\frac{P_0}{10\alpha}} \exp\Big(\frac{1}{2}\Big(\sigma_i\frac{\ln 10}{10\alpha}\Big)^2 - \frac{\ln 10}{10\alpha}z_i\Big).\end{aligned} \tag{2.26}$$

Substituting (2.9) into (2.26) gives [42]

$$\mathbb{E}[\hat{d}_{\mathrm{ML}_i}] = 10^{\frac{\ln 10}{2}(\frac{\sigma_i}{10\alpha})^2} d_i. \tag{2.27}$$

Obviously,

$$\mathbb{E}[\hat{d}_{\mathrm{ML}_i}] - d_i \neq 0 \quad \text{if } \sigma \neq 0. \tag{2.28}$$

Thus, the distance estimate (2.24) is biased. As shown in (2.27), the limit of $E[\hat{d}_{\mathrm{ML}_i}]$ is $d_i$ as $\sigma_i^2$ approaches $0$. Thus, when the shadowing noise is small enough, the bias is negligible. Moreover, the bias is proportional to the true distance. This may lead to inaccurate localization when the target is far away from the node.

The second moment of $\hat{d}_{\mathrm{ML}_i}$ can be computed in a similar way as for the expectation:

$$\begin{aligned}\mathbb{E}[(\hat{d}_{\mathrm{ML}_i}(P_i))^2] &= \int (\hat{d}_{\mathrm{ML}_i})^2 p(P_i; d_i)\,\mathrm{d}P_i \\ &= \int 10^{\frac{2(P_0-P_i)}{10\alpha}} \frac{1}{\sqrt{2\pi\sigma_i}} \exp\Big(-\frac{1}{2}\Big(\frac{P_i - \mathbf{z}_i}{\sigma_i}\Big)^2\Big)\,\mathrm{d}P_i \\ &= 10^{2\ln 10(\frac{\sigma_i}{10\alpha})^2} d_i^2.\end{aligned} \tag{2.29}$$

Further, the variance of $\hat{d}_{\text{ML}_i}$ is:

$$\begin{aligned}\text{Var}[\hat{d}_{\text{ML}_i}] &= \mathbb{E}[(\hat{d}_{\text{ML}_i} - \mathbb{E}[\hat{d}_{\text{ML}_i}])^2] \\ &= \mathbb{E}[(\hat{d}_{\text{ML}_i})^2] - (\mathbb{E}[\hat{d}_{\text{ML}_i}])^2 \\ &= \big(10^{2\ln 10(\frac{\sigma}{10\alpha})^2} - 10^{\ln 10(\frac{\sigma}{10\alpha})^2}\big) d_i^2 \\ &= \Big(\exp\Big(\frac{2(\ln 10)^2\sigma^2}{100\alpha^2}\Big) - \exp\Big(\frac{(\ln 10)^2\sigma^2}{100\alpha^2}\Big)\Big) d_i^2.\end{aligned} \tag{2.30}$$

The variance of the distance estimate is proportional to the true squared distance to be estimated. This means the distances measured at nodes near the target have smaller variance than nodes which are far away from the target.

Since the biased $\hat{d}_{\text{ML}_i}$ will lead to inaccurate location estimation, an unbiased estimate $\hat{d}_i$ is necessary. It is proved in [42], the only unbiased estimate of $d_i$ is

$$\hat{d}_i = \exp\Big(-\frac{0.1\ln 10(P_i - P_0)}{\alpha} - \frac{0.01(\ln 10)\sigma_i^2}{2\alpha^2}\Big). \tag{2.31}$$

The corresponding variance of the unbiased $\hat{d}_i$ is

$$\text{Var}(\hat{d}_i) = d_i^2\Big(\exp\frac{0.01(\ln 10)^2\sigma_i^2}{\alpha^2} - 1\Big). \tag{2.32}$$

where $\text{Var}(\cdot)$ represents the variance. Since $\hat{d}_i$ is unbiased, the mean-squared error is $\text{Var}(\hat{d}_i)$. For the biased estimator $\hat{d}_{\text{ML}_i}$, the mean-squared error is $\text{Var}(\hat{d}_{\text{ML}_i})$ plus its bias-squared. The difference between $\text{Var}(\hat{d}_{\text{ML}_i})$ and $\text{Var}(\hat{d}_i)$ can be expressed as:

$$\begin{aligned}\text{Var}[\hat{d}_{\text{ML}_i}] - \text{Var}[\hat{d}_i] &= \Big(\exp\Big(\frac{2(\ln 10)^2\sigma^2}{100\alpha^2}\Big) - \exp\Big(\frac{(\ln 10)^2\sigma^2}{100\alpha^2}\Big)\Big) d_i^2 - \Big(\exp\frac{0.01(\ln 10)^2\sigma_i^2}{\alpha^2} - 1\Big) d_i^2 \\ &= \Big(\exp\Big(\frac{2(\ln 10)^2\sigma^2}{100\alpha^2}\Big) + 1\Big) d_i^2 > 0.\end{aligned} \tag{2.33}$$

Therefore, $\hat{d}_i$ is better than $\hat{d}_{\text{ML}_i}$ in the sense of mean-squared error. When the distance estimate is available, a range-based method can be used to solve the problem. The variance of $\hat{d}_i$ could be further used to analyse the performance of the range based localization algorithms [16], [43]. The localization problem now becomes:

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \sum_{i=1}^{N} (\hat{d}_i - \parallel \mathbf{u} - \mathbf{c}_i \parallel)^2, \tag{2.34}$$

where $||\cdot||$ represents the norm operator. Solving the non-linear problem (2.34) requires numerical search methods such as the steepest descent or the Gauss-Newton method.

#### 2.1.3.2 Linear Least Squares Formulation

Besides the computational costly methods discussed above, Linear Least Squares (LLS) is the most widely used in distance based localization problem. Here, we introduce two well known LLS algorithms [17], [43].

Expanding (2.5), rearranging the equation in a matrix form, the LS is given as [43]:

$$\mathbf{A}\boldsymbol{\vartheta} = \mathbf{b} \tag{2.35}$$

where

$$\mathbf{A} = \begin{bmatrix} -2x_1 & -2y_1 & 1 \\ \vdots & \vdots & \vdots \\ -2x_N & -2y_N & 1 \end{bmatrix},$$

$$\boldsymbol{\vartheta} = \begin{bmatrix} x \\ y \\ x^2 + y^2 \end{bmatrix}^T$$

and

$$\mathbf{b} = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 \\ \vdots \\ d_N^2 - x_N^2 - y_N^2 \end{bmatrix}.$$

When there are less than 3 nodes in the network to locate the target, or all the nodes are placed in a straight line, $\mathbf{A}$ is rank deficient. In this case, there is not enough data to uniquely determine $\boldsymbol{\vartheta}$. In fact, there are at least two approximate solutions that reproduce the measurements. More measurements are required to decide which one is better.

When there are more than 3 nodes in the network to locating the target, and at least 3 nodes are not placed in a straight line, the system is overdetermined and $\mathbf{A}$ has full rank. In this case, an exact fit to the measurements is not possible. An approximate solution for $\mathbf{A}\hat{\boldsymbol{\vartheta}} \approx \hat{\mathbf{b}}$ may be calculated as (call it LLS-1):

$$\hat{\boldsymbol{\vartheta}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\hat{\mathbf{b}}. \tag{2.36}$$

where $\hat{\mathbf{b}}$ is the matrix $\mathbf{b}$ with $d_i^2$ replaced by $\hat{d_i^2}$. Then the location estimate is the first two variables of $\hat{\boldsymbol{\vartheta}}$. In this case, $\mathbf{A}$ has full rank, so $\mathbf{A}^T\mathbf{A}$ is nonsingular, and the solution for $\hat{\boldsymbol{\vartheta}}$ is unique. If $\hat{\mathbf{b}}$ is unbiased, then $\hat{\boldsymbol{\vartheta}}$ is unbiased.

When there are exact 3 nodes in the network and they are not placed in a straight line, the system is determined. The solution can also be obtained using (2.36).

In the literature, there are also other algorithms that make use of the relation between $\mathbf{u} = [\vartheta_1, \vartheta_2]^T$ and the third variable of $\boldsymbol{\vartheta}$. The LLS is calculated under the assumption that the elements in $\boldsymbol{\vartheta}$ are independent. With the fact that $\vartheta_3 = (\vartheta_1)^2 + (\vartheta_2)^2$, a constraint can be formulated based on it. In [44], the constraint is formulated as a LS with a Lagrange function. Correspondingly, the performance improves compared with (2.36).

An alternative LS can be formulated as follows [17] (Call it LLS-2). Fixing expression for the $r$th node in (2.5), substitute it from the rest of the equations for $i = 1, 2, \ldots, N(i \neq r)$. Note that $(x_r, y_r)$ is the coordinate of the $r$th node, and the quadratic term $x^2 + y^2$ is removed. Rearranging the equations, an alternative LS solution is

$$\hat{\mathbf{u}} = (\tilde{\mathbf{A}}^T \tilde{\mathbf{A}})^{-1} \tilde{\mathbf{A}}^T \tilde{\mathbf{b}}, \tag{2.37}$$

where

$$\tilde{\mathbf{A}} = \begin{bmatrix} x_1 - x_r & y_1 - y_r \\ \vdots & \vdots \\ x_N - x_r & y_N - y_r \end{bmatrix},$$

and

$$\tilde{\mathbf{b}} = \begin{bmatrix} \hat{d}_r^2 - \hat{d}_1^2 - x_r^2 - y_r^2 + x_1^2 + y_1^2 \\ \vdots \\ \hat{d}_r^2 - \hat{d}_N^2 - x_r^2 - y_r^2 + x_N^2 + y_N^2 \end{bmatrix}.$$

In order to get more insight about the accuracy of LLS-2 estimator, it is worth to analyze the error in the vector $\tilde{\mathbf{b}}$. Assuming that $\hat{d}_i$ is unbiased, then $\hat{d}_i$ is modeled as

$$\hat{d}_i = d_i + n_{d_i}, \tag{2.38}$$

where $n_{d_i}$ represents the zero-mean noise in $\hat{d}_i$. Then, the $i$th element of $\tilde{\mathbf{b}}$ with noise can be expanded as:

$$\begin{aligned} \tilde{\mathbf{b}}_i =& d_r^2 - d_i^2 - x_r^2 - y_r^2 + x_i^2 + y_i^2 \\ &+ 2d_r n_{d_r} - 2d_i n_{d_i} + n_{d_r}^2 - n_{d_i}^2. \end{aligned} \tag{2.39}$$

In a similar way, $\hat{\mathbf{b}}_i$ can be expanded as:

$$\hat{\mathbf{b}}_i = d_i^2 - x_i^2 - y_i^2 - 2d_i n_{d_i} + n_{d_i}^2. \tag{2.40}$$

Compared (2.40) with (2.39), while the quadratic term $x^2 + y^2$ is discarded, the number of noise terms in $\tilde{\mathbf{b}}$ increases. In particular, the noise term depends on the distance between the reference node and the target. Thus, the performance of the LLS-2 degrades as the target moves away from the reference node. Both the two methods introduced have low computational complexity for the nature of LS methods.

#### 2.1.3.3 Semidefinite Programming Formulation

When the estimate of distance $\hat{d}_i$ is available, another way to estimate the target position is to use convex optimization method. Expanding $\hat{d}_i$ using (2.5), after some manipulation, (2.34) can be written as:

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \sum_{i=1}^{N} (\mathbf{u}^T\mathbf{u} - 2\mathbf{c}_i^T\mathbf{u} + x_i^2 + y_i^2 - \hat{d}_i^2). \tag{2.41}$$

A possible way to transfer (2.41) into a convex problem is to use semidefinite relaxation technique. Let $\bar{\mathbf{U}} = \mathbf{u}\mathbf{u}^T$, then $\mathbf{u}^T\mathbf{u} = \mathrm{Tr}(\bar{\mathbf{U}})$, where $\mathrm{Tr}(\cdot)$ represents the trace operator. (2.41) can be further written as

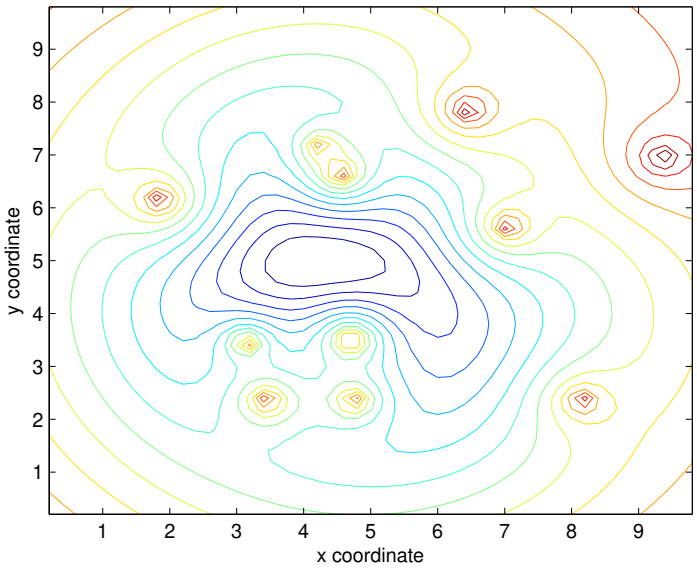
$$\begin{aligned} \underset{\bar{\mathbf{U}},\mathbf{u}}{\text{minimize}} \quad & \sum_{i=1}^{N} (\mathrm{Tr}(\bar{\mathbf{U}}) - 2\mathbf{c}_i^T\mathbf{u} + x_i^2 + y_i^2 - \hat{d}_i^2)^2 \\ \text{subject to} \quad & \bar{\mathbf{U}} - \mathbf{u}\mathbf{u}^T = 0 \end{aligned} \tag{2.42}$$

The objective function is now convex, but the constraint is neither semidefinite nor linear. The equality constraint, $\bar{\mathbf{U}} - \mathbf{u}\mathbf{u}^T = 0$, can be relaxed by replacing it with a semidefinite constraint, $\bar{\mathbf{U}} - \mathbf{u}\mathbf{u}^T \succeq 0$. Here, $\succeq$ represents the matrix on the left side of $\succeq$ is a positive semidefinite matrix. Using the Schur complement, the constraint is equal to

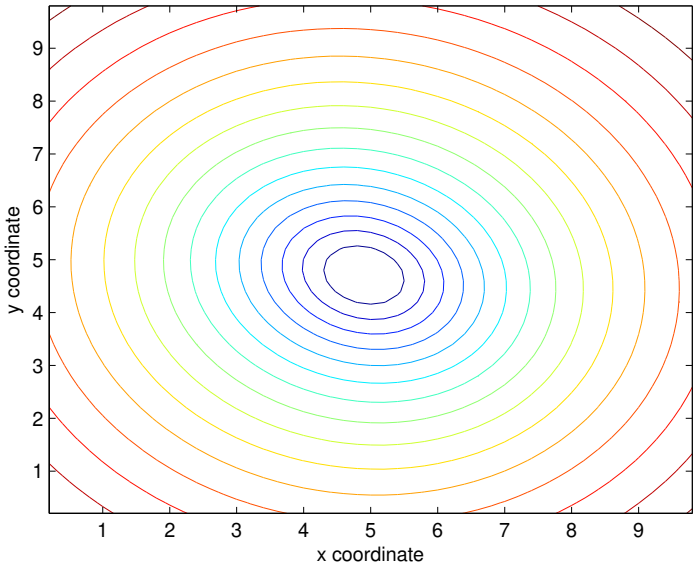
$$\begin{bmatrix} \bar{\mathbf{U}} & \mathbf{u} \\ \mathbf{u}^T & 1 \end{bmatrix} \succeq 0 \tag{2.43}$$

Now, the objective function is convex, and the constraint is semidefinite. Then this problem can be solved using convex optimization tool box, i.e., CVX [45]. The contour maps of the objective function (2.42) and the ML estimator (2.12) are plotted in Fig.2.2 for comparison. In

the simulation, eleven nodes are placed at fixed positions (3.5,2.4), (4.8,2.4), (4.7,3.5), (3.2,3.5), (7.1,5.7), (4.6,6.7), (6.5,7.9), (9.4,7), (8.2,2.4), (4.3,7.2) and (1.9,6.2), all in meters. The target position is (4,5)m, the transmit power is 0dBm, the standard deviation of noise is $\sigma = 3$dB and the PLE is 3.




(a) The ML cost function (2.12).




(b) The modified cost function (2.42) for SDP.

Figure 2.2: Contour plot of the objective function.

An alternative way to formulate SDP is to modify the original signal model (2.4) directly. Rearranging (2.4) gives

$$10^{(P_i-P_0)/(10\alpha)}d_i = 10^{n_i/(10\alpha)}. \tag{2.44}$$

By using a Taylor series expansion, the right hand-side of (2.44) can be approximated as

$$10^{(P_i-P_0)/(10\alpha)}d_i = 1 + \frac{\ln 10}{10\alpha}n_i. \tag{2.45}$$

After some manipulation, the location can be estimated by using a weighted nonlinear least squares (NLS) [46]:

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}}(\bar{\mathbf{L}}\mathbf{d} - \mathbf{1}_N)^T\mathbf{Q}_s^{-1}(\bar{\mathbf{L}}\mathbf{d} - \mathbf{1}_N). \tag{2.46}$$

where

$$\bar{\mathbf{L}} = \text{diag}\{10^{(P_1-P_0)/(10\alpha)}, \ldots, 10^{(P_N-P_0)/(10\alpha)}\}, \tag{2.47}$$

$$\mathbf{d} = [d_1, \ldots, d_N]^T, \tag{2.48}$$

$$\mathbf{Q}_s = (\ln 10)^2/(10\alpha)^2\mathbf{Q} \tag{2.49}$$

and $\mathbf{1}_N$ is a length $N$ all one vector. The cost function (2.46) is smoother than the original ML cost function (2.12). Thus, it can be easily relaxed into an SDP problem.

In [18], another SDP method is formulated. The Path Loss model (2.3) is modified, and a new ML-type cost function with no logarithm in the residual is formulated. Then the new cost function is relaxed into SDP problems. It has relatively low performance compared newly proposed SDP methods, i.e., [23]. In general, the performance of SDP methods depends on the error introduced during the approximation from the original cost function (2.12) to the convex cost function and the relaxation of the constraint. And the computational cost of SDP methods is way larger than the computational cost of LS.

Besides LS and SDP, in [47], the RSS based localization problem is formulated as a generalized

trust region subproblem (GTRS) [48]. Similar to (2.45), the original signal model is reformulated. Under the transformed model, the unscented transform (UT) [49] is employed to obtain the corresponding means and variances of the transformed measurements, and thus obtain a ML-type cost function. Further, the cost function is optimised as a GTRS using bisection method. The two methods above provide a less computational complexity than the SDP with a similar performance.

Some of the well-cited algorithms introduced above will be compared with respect to localization accuracy and computational cost in Chapter 3.

## 2.2 Unknown Channel Parameters

In practical applications, the channel parameters, i.e. the transmit power $P_0$ and the PLE $\alpha$, are typically not known *a priori*. To obtain accurate estimates on these parameters, a lot of training data are required in the calibration process, which is not practical in some scenarios. Further, due to battery exhaustion, the transmit power may also be time varying. Due to change of weather, change of landscape and human behaviours, the PLE may be time varying too. Thus, the system need to be frequently re-calibrated, resulting in hardware and software consumption. This section reviews existing localisation techniques when transmit power or PLE or both of them is unknown to the system.

### 2.2.1 Unknown Transmit Power

In the case of unknown transmit power, the techniques used to estimate the target location can be classified into two categories: the joint estimate of location and transmit power and the use of differential received signal strength (DRSS). This section reviews some existing algorithms and introduces the DRSS signal model.

#### 2.2.1.1 Jointly estimation techniques

Similar to the maximum likelihood estimator derived in (2.12), the ML estimator for the joint estimate problem is

$$\hat{\boldsymbol{\theta}}_{\text{ML}} = \arg\min_{\boldsymbol{\theta}} (P_i - P_0 + 10\alpha \log_{10} d_i)^2 \tag{2.50}$$

where $\hat{\boldsymbol{\theta}}_{\text{ML}} = [x, y, P_0]^T$.

The existing localisation techniques for the jointly estimation of transmit power and location is actually similar to the methods introduced in previous section. In [22], the Path Loss model (2.4) is reformulated using Taylor Series Expansion as with the process in (2.44)-(2.45). By using the reformulated signal model, a new cost function can be built as in (2.46). Then the cost function can be relaxed into SDP problems similar to (2.46). Alternatively, the new cost function can also be linearized to formulate a LS function [23]. Then the two-step Weighted Least Squares (WLS) [50] is employed to solve the LS function. In [51], similar to the process in [22], the original signal model is transformed into a NLS function using Taylor Series Expansion. Further, the NLS is expressed as a quadratic program which is a Generalized Trust Region Subproblem (GTRS). Then a bisection is recruited to give a solution. Another GTRS-type based method is given in [52]. The original signal model is reformulated like (2.45). A new NLS cost function is formulated. Then the UT is used to calculate the mean and variance for the NLS cost function. Finally, the cost function is solved by a bisection method. The aforementioned methods are actually joint estimation versions of the methods reviewed in Section 2.1.3. Their merits and drawbacks are exactly the same as their original versions. The SDP method [22] has heavy computational cost compared with other methods. The GTRS algorithms [51], [52] have slightly better performance than SDP with much less computational cost. However, in reality, especially in indoor environments, the shadowing noise are severely correlated. The methods described above and their original versions can not easily be adapted to make use of the correlated shadowing noise.

#### 2.2.1.2 Differential Received Signal Strength

A different approach for estimating the target position when the transmit power $P_0$ is unknown is to eliminate $P_0$ from the RSS model. Using one RSS measurement to subtract another, the DRSS measurement is defined as:

$$q_{ij} = P_i - P_j = -10\alpha \log_{10} r_{ij} + n_{v_{ij}}, \quad 1 \leq i, j \leq N, i \neq j. \tag{2.51}$$

where $n_{v_{ij}} = n_i - n_j$, and $r_{ij} = \frac{d_i}{d_j}$, is the ratio of distance $d_i$ and distance $d_j$.

The set of measurements is divided into $N-1$ basic and $(N-1)^2$ redundant measurements. The redundant measurements can be determined by a linear combination of the basic measurements.

To formulate a ML estimator, let $j$ be a fixed integer in the range $1 \leq i, j \leq N$ and consider the $N-1$ basic DRSS measurements. For simplicity, denote $j$ as 1, and only the following measurements are considered:

$$q_{i1} = P_i - P_1 = -10\alpha \log_{10} r_{i1} + n_{v_{i1}}, \quad i = 2, \ldots, N. \tag{2.52}$$

Denote the DRSS measurements vector as

$$\mathbf{q} = [q_{21}, q_{31}, \ldots, q_{N1}]^T, \tag{2.53}$$

though the RSS measurements are uncorrelated, the DRSS values become correlated.

$$\mathbf{Q}_v = E\{\mathbf{q}\mathbf{q}^T\} = \sigma^2(\mathbf{I}_{N-1} + \mathbf{1}_{N-1}\mathbf{1}_{N-1}^T). \tag{2.54}$$

Thus,

$$\mathbf{q} \sim \mathcal{N}(E\{\mathbf{q}\}, \mathbf{Q}_v). \tag{2.55}$$

And the corresponding ML estimator for DRSS model is

$$\hat{\mathbf{u}}_{\text{ML}} = \arg\min_{\hat{\mathbf{u}}} (\mathbf{q} - E\{\mathbf{q}\})^T \mathbf{Q}_v^{-1} (\mathbf{q} - E\{\mathbf{q}\}). \tag{2.56}$$

The corresponding FIM derived in [53] is

$$\mathbf{J}_D = \frac{100\alpha^2}{(\ln 10)^2} \mathbf{\Omega}^T \mathbf{Q}_v^{-1} \mathbf{\Omega}, \tag{2.57}$$

where

$$\mathbf{\Omega} = \begin{bmatrix} \frac{x-x_2}{d_2^2} - \frac{x-x_1}{d_1^2} & \frac{y-y_2}{d_2^2} - \frac{y-y_1}{d_1^2} \\ \vdots & \vdots \\ \frac{x-x_N}{d_N^2} - \frac{x-x_1}{d_1^2} & \frac{y-y_N}{d_N^2} - \frac{y-y_1}{d_1^2} \end{bmatrix}. \tag{2.58}$$

The CRLB is the inverse of the FIM. It is proved in [54], that the CRLB of DRSS based localization is the same as the CRLB of RSS based localization with unknown transmit power.

Next, we use an example to explain the geometry of DRSS based localization. Consider a case where one target is at $\mathbf{u} = [x, y]^T$, and two sensor nodes at $\mathbf{c}_1 = [x_1, y_1]^T$, $\mathbf{c}_2 = [x_2, y_2]^T$. Denote the distance between $\mathbf{c}_1$ and $\mathbf{c}_2$ as $d_{12}$. Substituting (2.5) into $r_{21} = d_2/d_1$ gives

$$(x - x_1)^2 + (y - y_1)^2 = r_{21}^2((x - x_2)^2 + (y - y_2)^2). \tag{2.59}$$

If $r_{21} = 1$, (2.59) is a linear equation

$$(x_1 - x_2)x + (y_1 - y_2)y = \frac{1}{2}(x_1^2 + y_1^2 - x_2^2 - y_2^2). \tag{2.60}$$

which represents the perpendicular bisector of the line passing through $\mathbf{c}_1$ and $\mathbf{c}_2$ [55].

If $r_{21} \neq 1$, (2.41) is a circle centred at

$$(\frac{x_2 - r_{21}x_1}{1 - r_{21}}, \frac{y_2 - r_{21}y_1}{1 - r_{21}}) \tag{2.61}$$

with a radius

$$\frac{\sqrt{r_{21}d_{12}}}{|1 - r_{21}|}. \tag{2.62}$$

As shown above, each measurement $q_{i1}$ in the DRSS measurement model represents a circle. Thus, the solution to the localization problem using DRSS model is, in fact, to find the intersection of multiple circles. As an illustration, 10 circles represented by 10 DRSS measurements without noise obtained by 5 nodes are plotted in Fig.2.3.

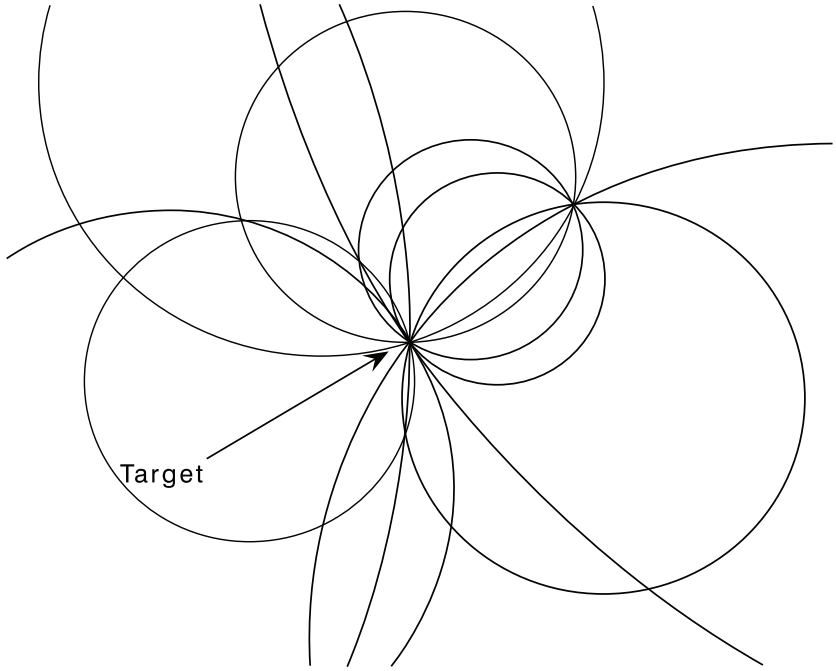



Figure 2.3: Target at the intersection of circles associated with 10 DRSS measurements.

The classic way to solve the DRSS localization problem is to use LS [29], [56]. This methods consists of 2 steps. In the first step, similar to the process from (2.21) to (2.24), the ML estimate of the distance ratio $r_{i1}$ is

$$\hat{r}_{i1} = 10^{-\frac{P_i - P_1}{10\alpha}}. \tag{2.63}$$

In the second step, the LS is formulated as follows. Expanding (2.59) yields

$$(r_{21}^2 - 1)(x^2 + y^2) + (2x_i - 2r_{i1}^2 x_1)x + (2y_i - 2r_{i1}^2 y_1)y + r_{i1}^2(x_1^2 + y_1^2) - (x_i^2 + y_i^2) = 0. \tag{2.64}$$

Replacing the true $r_{i1}$ by $r_{\mathrm{ML}_{21}}$, the solution of LS is

$$\hat{\boldsymbol{\vartheta}} = (\breve{\mathbf{A}}^T \breve{\mathbf{A}})^{-1} \breve{\mathbf{A}}^T \breve{\mathbf{b}}, \tag{2.65}$$

where

$$\breve{\mathbf{A}} = \begin{bmatrix} (2x_2 - 2\hat{r}_{21}^2 x_1) & (2y_2 - 2\hat{r}_{21}^2 y_1) & \hat{r}_{21}^2 - 1 \\ \vdots & \vdots & \vdots \\ (2x_2 - 2\hat{r}_{N1}^2 x_1) & (2y_2 - 2\hat{r}_{N1}^2 y_1) & \hat{r}_{N1}^2 - 1 \end{bmatrix},$$

and

$$\breve{\mathbf{b}} = \begin{bmatrix} \hat{r}_{21}^2(x_1^2 + y_1^2) - (x_2^2 - y_2^2) \\ \vdots \\ \hat{r}_{N1}^2(x_1^2 + y_1^2) - (x_N^2 - y_N^2) \end{bmatrix}$$

Note that $\hat{\boldsymbol{\vartheta}}$ is the estimate of $\boldsymbol{\vartheta}$ which is defined in (2.35). The ordinary LS above has lower computational complexity and lower localization accuracy compared with the SDP proposed in [57]. Its performance could be further improved by using an unbiased estimator $\hat{r}_{i1}$ and formatting a weighted LS with the covariance of $\hat{r}_{i1}$.

### 2.2.2 Unknown Path Loss Exponent

In the case of unknown PLE, The joint ML estimation of the target $\mathbf{u}$ and the PLE $\alpha$ when $P_0$ is known can be written as

$$\hat{\boldsymbol{\theta}}_{\text{ML}} = \arg\min_{\boldsymbol{\theta}} (P_i - P_0 + 10\alpha \log_{10} d_i)^2 \tag{2.66}$$

where $\hat{\boldsymbol{\theta}}_{\text{ML}} = [x, y, \alpha]^T$.

#### 2.2.2.1 Deterministic PLE

One straight way to jointly estimate the target location and the PLE is to employ iterative methods to solve the cost function. In [58], the solution of the NLS is obtained using the LevenbergMarquardt method which is a modification to the GN method. In [25] and [59], grid search is used to give a solution to the problem.

Another approach is based on the derivations in Section 2.1.3 [23], [52]. Note that if an initial of the PLE $\hat{\alpha}$ is available, then the target location can be calculated using the methods introduced in Section 2.1.3. When an estimate of $\mathbf{u}$ is obtained, (2.4) is linear with respect to $\alpha$. This makes it possible to alternatively estimate $\hat{\mathbf{u}}$ and $\alpha$. The estimation procedure is shown as follows

1. Obtain an initial estimate of $\alpha$, $\hat{\alpha}(0)$, then estimate $\hat{\mathbf{u}}(1)$ using methods introduced in Section 2.1.3, i.e., (2.36), (2.42).

2. Using $\hat{\mathbf{u}}(t)$, where $t$ represents the index of iteration, $\alpha$ is estimated as:

$$\begin{aligned}\hat{\alpha}(t) &= \arg\min_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \frac{(P_i - P_0 + 10\alpha \log_{10} \parallel \hat{\mathbf{u}}(t-1) - \mathbf{c}_i \parallel))^2}{\sigma_i^2} \\ &= \frac{\sum_{i=1}^{N} [10\alpha \log_{10}(\parallel \hat{\mathbf{u}}(t-1) - \mathbf{c}_i \parallel)(P_i - P_0)]/\sigma_i^2}{\sum_{i=1}^{N} (10 \log_{10}(\parallel \hat{\mathbf{u}}(t-1) - \mathbf{c}_i \parallel))^2/\sigma_i^2}.\end{aligned} \tag{2.67}$$

3. Use the update $\hat{\alpha}(t)$ to obtain an updated estimate $\hat{\mathbf{u}}(t)$ with methods introduced in Section 2.1.3, i.e., (2.36), (2.42), then go to step 2. Iteration stops when $\hat{\alpha}$ converges or the maximum iteration is reached.

The initial value of $\alpha$ can be chosen based on the empirical values in [34].

#### 2.2.2.2 Random PLE

A research on path loss model is carried out in [28]. The RSS measurements are collected in 23 homes. The homes have different structure, age, size and clutter. The transmitter is located in a fixed position, and the receiver is moved throughout the homes on a pre-measured grid. The data shows that the values of $\alpha$ change from one home to another and have a Gaussian distribution. In [26], the PLE $\alpha$ is considered as a random variable that is Gaussian distributed with a mean $\mu_\alpha$ and variance $\sigma_\alpha$. With the random assumption of $\alpha$, the maximum a posterior (MAP) estimator of $\hat{\boldsymbol{\theta}} = [x, y, \alpha]^T$ is given by [39]

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \{\ln p(\boldsymbol{\theta}|\mathbf{p})\}. \tag{2.68}$$

Since in $\boldsymbol{\theta}$, $\alpha$ is a random variable, thus

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\alpha}\{\ln p(\mathbf{p}|\alpha, \mathbf{u}) + \ln p(\alpha)\}, \tag{2.69}$$

where $p(\alpha)$ is the pdf of $\alpha$. The MAP solution is given by

$$\hat{\alpha}_{\text{MAP}}(t) = \mu_\alpha + \frac{\sum_{i=1}^{N}((\frac{P_i - P_0}{\sigma_i}) - \mu_\alpha(10\log_{10}(\parallel \hat{\mathbf{u}}(t-1) - \mathbf{c}_i \parallel))/\sigma_i^2}{1/\sigma_\alpha^2 + \sum_{i=1}^{N} 10\log_{10}(\parallel \hat{\mathbf{u}}(t-1) - \mathbf{c}_i \parallel)/\sigma_i^2} \tag{2.70}$$

The estimate process of random PLE is the same as the deterministic PLE, except (2.67) in step 2 is replaced by (2.70). The use of MAP requires acknowledgement of the statistical information of $\alpha$, so it is only used in this model. In most cases, the MAP has better performance due to the additional information $\sigma_\alpha$ available. However, when the target is at the center of the sensor node geometry, the two estimators have similar performance [26]. The MAP method should be a better choice if statistical information of $\alpha$ is available.

Notice that whatever modeling method for PLE is used, the methods described above requires an initialization. And the initial may influence the final estimate. As (2.66) is also multimodal, techniques without initial and having the ability to handle multimodal optimization may be applied to solve this problem.

Until now, the methods used in RSS based localization problem without transmit power or PLE have been reviewed. A more practical problem is that to locate the target while both PLE and transmit power are unknown to the system. As can be seen, the above methods used in joint PLE estimation are iterative methods. Therefore, these methods could be easily combined with joint transmit power estimation methods or DRSS-based estimation methods. In [57], DRSS based localization method is combined with the search methods introduced above. In [23], [52], [60], joint transmit power estimation methods are combined with the search methods described above. These methods share the same drawbacks with the unknown PLE localization problem.

## 2.3 Quantized Received Signal Strength

In WSNs, the sensor nodes are characterized by limited resources such as energy and bandwidth, One way to save energy is to limit the data transmitted within the network. It is desirable that only quantized multibit data is transmitted within the network. In this section, we introduce the basic concept of quantized received signal strength. To better understand the property of QRSS,

the channel parameters $P_0$, $\alpha$, $\sigma$ are assumed to be known.

### 2.3.1 Proximity Measurement

RSS and proximity measurements are widely used in positioning when cheap devices are deployed in WSNs. The proximity measurement is binary data which is 1 if a packet transmitted by a device can be successfully decoded by the device that receives the packet. Otherwise, the proximity measurement is 0. In other words, two devices are in the range of communication if the proximity measurement between them is one. One big advantage of proximity measurement is that it does not need additional communications cost to measure the proximity. Proximity measurements based localization problem has been widely studied by researchers [61], [62], [63], [64].

For simplicity, the proximity measurement received at $i$th node is denoted as $K_i$. If the received signal strength $P_i$ is larger than a threshold $s_1$, the node $i$ is connected with the target. Therefore,

$$K_i = \begin{cases} 1 & P_i \geq s_1 \\ 0 & P_i < s_1 \end{cases}. \tag{2.71}$$

Given the path loss model (2.4) and the proximity measurement (2.71), the probability mass function (PMF) of the received proximity measurement $K_i$ is [32]

$$p(K_i = m|\mathbf{x}) = m + (-1)^m Q(\bar{\xi}_i(1)), \tag{2.72}$$

$$\bar{\xi}_i(m) = \frac{10\alpha \log_{10} \frac{\|\mathbf{x}-\mathbf{c}_i\|}{d_m}}{\sigma}, \tag{2.73}$$

where $m \in \{0, 1\}$ and $d_m$ is the distance where the received signal strength $P_i$ is equal to the threshold $s_1$:

$$d_m = 10^{\frac{P_0 - s_1}{10\alpha}}. \tag{2.74}$$

And $Q(\cdot)$ is the cumulative density function (CDF) of a univariate zero-mean unit-variance Normal distribution

$$Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} \exp(-\frac{t^2}{2})dt. \tag{2.75}$$

### 2.3.2 Quantized Received Signal Strength

The Quantized RSS measurement is an extension of proximity measurements. The proximity measurement is one-bit data determined by one threshold. In QRSS, the RSS measurement is quantized into a multi-bit data determined by a series of thresholds. The relationship between QRSS measurement and RSS measurement is obvious. As the number of bits used in QRSS increases, the effort of quantization decreases. In reality, signals are always quantized; hence, so is RSS. With enough number of bits used in QRSS measurement, QRSS measurement is the same as the RSS measurement.

One advantage of using QRSS measurement is that it can simplify the design of an analog-to-digital converter. When an analog-to-digital converter is used to quantize the raw (analog) received signal strength, the complexity of the converter increases with the number of quantization threshold.

Extending the one-bit proximity measurements model into a multi-bit QRSS measurement model, it is assumed that $M$ bits ($L = 2^M - 1$ levels) data are used. The quantization threshold for analog received signal strength is denoted as $\mathbf{s} = [s_0, s_1, \ldots, s_L]$. Thus, the QRSS measurement $K_i$ is calculated as:

$$K_i = \begin{cases} 0 & \text{if } s_0 < P_i < s_1 \\ 1 & \text{if } s_1 < P_i < s_2 \\ \vdots & \vdots \\ L-1 & \text{if } s_{L-1} < P_i < s_L. \end{cases} \tag{2.76}$$

Since, there is no upper bound or lower bound for the received signal strength $P_i$. $s_0$ and $s_L$ are defined as $-\infty$dBm and $\infty$dBm respectively.

Given (2.72) and (2.76), the PMF of QRSS measurement model when $M \geq 2$ can be expressed as

$$p(K_i = m|\mathbf{x}) = Q(\bar{\xi}_i(m+1)) - Q(\bar{\xi}_i(m)) \tag{2.77}$$

where $\bar{\xi}_i(m)$ is defined in (2.73) with $m = 1, 2, \ldots, L-1$. When $M = 1$, the multi-bit QRSS measurement is simplified to the proximity measurement as shown in (2.72).

### 2.3.3 Optimal Threshold Design Method

The localization accuracy depends on the information contained in the QRSS measurement. And the information in QRSS measurement is determined by a threshold.

An optimal threshold design method is described in [32]. Here, we give an example to show how this method works. Consider a 10m by 10m area with four nodes placed at [0,0], [0,10], [10,0] and [10,10], all in metres. And the target is located at [4,5]m. Furthermore, we assume the transmit power is $-10$dBm, the PLE and standard deviation of shadowing noise is 3 and 6dB respectively. In addition, three-level quantization is employed, $\mathbf{s} = [s_0, s_1, s_2, s_3]^T$, where $s_0$ and $s_3$ is $-\infty$dBm and $\infty$dBm respectively. Obviously, if $s_1$ is too small or $s_2$ is too large , the 3-level QRSS is simplified to proximity measurement. The localization variance is maximized if the CRLB of the QRSS based localization is minimized. Thus, maximizing the FIM will give the optimal thresholds. The FIM for QRSS based localization is derived in [32]. It is a $2 \times 2$ matrix with elements:

$$\begin{aligned}
\mathbf{J}_{11} &= 50\alpha^2 \sum_i \xi_i (x - x_i)^2 / ((\log 10)^2 \pi d_i^4 \sigma^2), \\
\mathbf{J}_{22} &= 50\alpha^2 \sum_i \xi_i (y - y_i)^2 / ((\log 10)^2 \pi d_i^4 \sigma^2), \\
\mathbf{J}_{12} &= \mathbf{J}_{21} = 50\alpha^2 \sum_i \xi_i (x - x_i)(y - y_i) / ((\log 10)^2 \pi d_i^4 \sigma^2),
\end{aligned} \tag{2.78}$$

where

$$\xi_i = \sum_{s=0}^{K-1} \frac{\exp(-(s_i - z_i)^2/2\sigma^2) - \exp(-(s_{i+1} - z_i)^2/2\sigma^2)}{Q(\frac{s_i - z_i}{\sigma}) - Q(\frac{s_{i+1} - z_i}{\sigma})}. \tag{2.79}$$

Fig.2.4 shows the FIM on $x$ when different value of $d_{m1}$ and $d_{m2}$ are used. The relationship between $d_{mi}$ and $s_i$ is shown in (2.74). Since $s_1$ is smaller than $s_2$, only half of the figure is shown.

As shown in Fig.2.4, the value of $d_{m1}$ and $d_{m2}$ that maximize the FIM is 4.8m and 8m. Correspondingly, the optimal threshold for the localization problem is -30.44dBm and -37.09dBm.

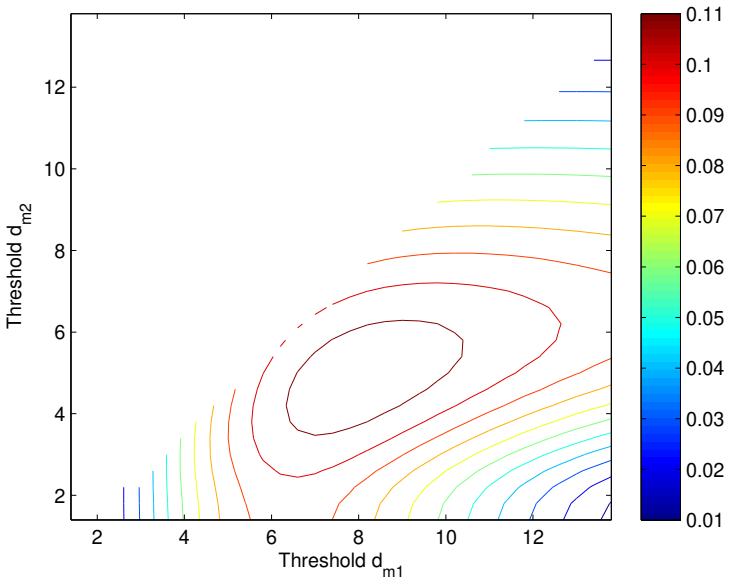



Figure 2.4: The contour map of Fisher information on $x$.

Similar to the 3-level case, the quantization threshold for multi-bit cases can be optimized by maximizing the FIM. A problem of the optimal threshold design method is that it requires the location of the target which is not applicable in real scenarios.

## 2.4 Conclusion

In this chapter, firstly the basics of the RSS signal model with some existing localization techniques are reviewed. Then localization techniques in the presence of unknown channel parameters are introduced. New localization methods will be proposed in Chapters 3 and 4, and their performance will be compared with existing methods. After the introduction of RSS models, we continue to demonstrate the quantization effects on RSS-based localization. As part of Chapter 5, the threshold design method for QRSS and corresponding localization methods are proposed. Eventually, some basic concepts of distributed optimization is introduced. With the help of these distributed techniques, the distributed algorithm for RSS- and QRSS-based localization problem is proposed in Chapters 4 and 5 respectively.

# Chapter 3

# Constrained Weighted Least Squares Algorithm Using RSS Measurements

## 3.1 Introduction

This chapter addresses the problem of target localization when the received signal strength (RSS) measurements are assumed to be correlated. The unknown position of the target is estimated by a maximum likelihood (ML) estimator. The resulting objective function, unfortunately, is complicated to solve because the measurements $P_i$ are highly non-linear and non-convex with respect to the unknown position $\mathbf{u}$. In Chapter 2, some classic ways to linearize or relax the objective function have been reviewed. However, the majority of existing studies in RSS based localization simply assume that shadowing noise at two positions is independent. In practice, shadowing noise is correlated due to similar topography among RSS propagation paths. The correlation coefficient is mainly affected by frequency, direction travelled and topography [27]. The data in [28] shows that as the angle and distance between a pair of locations decreases, the correlation tends to increase. Thus, it is very important to take into account the correlation for RSS-based location estimation. To solve the RSS localization problem with correlated shadowing, first, we extend the unbiased distance estimate based on uncorrelated noise to a correlated noise. Then, with the unbiased distance estimate, a constrained weighted least squares (CWLS) is formulated. Further, an iterative technique based on Newton's method is employed to give a solution. The CWLS method is also extended to be used in the differential RSS (DRSS) model in which transmit power is unknown.

The rest of the chapter is organised as follows. In Section 3.2, the RSS signal model with correlation is formulated. In Section 3.3, the CWLS method is developed. The computational complexity is analysed in Section 3.4. In Section 3.5, we discuss the simulation results. Concluding remarks are given in Section 3.6.

## 3.2 Problem Formulation

The RSS model with correlated shadowing noise will be described in this section. Assume there is a 2-dimensional wireless sensor network (WSN) that has a target with known coordinates denoted by $\mathbf{u} = [x, y]^T$. The WSN locates the target using $N$ nodes with known position denoted by $\mathbf{c}_i = [x_i, y_i]^T$, $i = 1, 2, \ldots, N$. The received signal strength $P_i$ is modelled as

$$P_i = P_0 - 10\alpha \log_{10} d_i + n_i, \quad i = 1, \ldots, N, \tag{3.1}$$

where $n_i$ follows a zero-mean Gaussian distribution shown in (2.3). However, (2.3) only describes the shadowing noise when the received signal strength is at a single position. The shadowing noise at two position are related by a bivariate Gaussian distribution as follows [65]:

$$f(n_i, n_j) = \frac{1}{2\pi\sigma_i\sigma_j\sqrt{(1-\rho^2)}} \exp\Big[\Big(-\frac{1}{2(1-\rho^2)}\Big)\Big(\frac{n_i^2}{\sigma_i^2} - \frac{2\rho n_i n_j}{\sigma_i\sigma_j} + \frac{n_j^2}{\sigma_j^2}\Big)\Big] \tag{3.2}$$

where $\rho$ is the correlation coefficient between the shadowing noise of the received signal strength at two positions. As mentioned in Chapter 2, $\alpha$ and $\sigma$ can be calculated using linear regression, the difference between $P_i$ and $10\alpha \log_{10} d_i$ being the shadowing noise. Then, the cross-correlation factor can be computed on the shadowing noise calculated. Further, the correlation coefficient will be the cross-correlation divide the variance $\sigma^2$ [27]. Note that the data used to estimate $\alpha$ and $\sigma$ can also be used to derive the correlation coefficient, thus no extra experiment is needed for calculating $\rho$. In the simulation section, we will also test the impact of inaccurate $\rho_{ij}$ on the performance of the localization algorithm. Similar to [29], [46] and [66], the covariance matrix of $n_i$ is assumed to be

$$\mathbf{Q}_{ij} = \begin{cases} \sigma_i^2 & \text{if } i = j \\ \rho_{ij}\sigma_i\sigma_j & \text{if } i \neq j, \end{cases} \tag{3.3}$$

where $\sigma_i$ is the standard deviation of shadowing noise and $\rho_{ij}$ is the correlation coefficient between the $i$th and $j$th RSS measurements. In the following, we assume $P_0, \alpha, \sigma_i$ and $\rho_{ij}$ are known to the system.

## 3.3 Algorithm Development

### 3.3.1 Known Transmit Power

As shown in Section 2.1.2, in most literature, least square (LS) or semidefinite programming (SDP) are employed to solve this problem. Here we choose the LS because it has lower computational cost than the SDP method which increases the battery life of the sensor nodes in WSN. To use the LS method, the first step is to estimate the distance between the target and nodes described in Section 2.1.2. As the unbiased estimator (2.31) is better than the biased one (2.24) in the sense of error covariance, here we choose the unbiased estimator. As shown in (2.35), to use the weighted LS, the covariance of $\hat{d}_i^2$ is necessary. The unbiased estimator and its covariance are derived in [42], explained as follows.

Multiplying both sides of the RSS signal model in (3.2) by $0.1(\ln 10)$ gives

$$0.1(\ln 10)P_i = 0.1(\ln 10)P_0 - \alpha \ln(d_i) + n_{r_i} \tag{3.4}$$

where $n_{r_i} = 0.1(\ln 10)n_i$, and the corresponding covariance is

$$[\mathbf{Q}_\omega]_{ij} = \begin{cases} 0.01(\ln 10)^2\sigma_i^2 & \text{if } i = j \\ 0.01(\ln 10)^2\rho_{ij}\sigma_i\sigma_j & \text{if } i \neq j \end{cases}. \tag{3.5}$$

Similar to the process from (2.21) to (2.24), the ML estimator for $d_i^2$ is

$$\begin{aligned} \hat{d}^2_{\text{ML}_i} &= 10^{\frac{P_0 - P_i}{5\alpha}} \\ &= \exp\Big(\frac{(\ln 10)(P_0 - P_i)}{5\alpha}\Big). \end{aligned} \tag{3.6}$$

Rearranging (3.4) gives

$$\exp\Big(\frac{\ln(10)(P_0 - P_i)}{5\alpha}\Big) = \exp\Big(-\frac{2n_{r_i}}{\alpha}\Big)d_i^2, \tag{3.7}$$

therefore, $\hat{d}^2_{\text{ML}_i}$ can also be expressed as

$$\hat{d}^2_{\mathrm{ML}_i} = \exp(-\frac{2n_{r_i}}{\alpha})d_i^2. \tag{3.8}$$

Denote $\hat{d}^2_{\mathrm{ML}_i}$ as $z_u$ for notation simplicity. Considering an arbitrary estimate $f(z)$ whose mean is $d_i^2$ for all $d_i^2 > 0$, i.e.,

$$\mathbb{E}[f(z_u)] = d_i^2. \tag{3.9}$$

As $\alpha$ and $\sigma_i$ are known, $f(\cdot)$ can be a function of $\alpha$ and $\sigma_i$. There holds

$$d_i^2 = \frac{1}{0.1\ln(10)\sqrt{2\pi}\sigma_i}\int_{-\infty}^{\infty} f\big(\exp\big(-\frac{2n_{r_i}}{\alpha}\big)d_i^2\big)\exp\Big(-\frac{n_{r_i}^2}{0.02(\ln 10)^2\sigma_i^2}\Big)\mathrm{d}n_{r_i}. \tag{3.10}$$

In [42], the authors substitute $z_u$ into (3.10) and uses several intermediate variables to transform (3.10) into a a form of Laplace transformation. Thus, the right hand side and left hand side of the equation is a Laplace pair. Then, the uniqueness of Laplace pair establishes:

$$f(z_u) = \exp\Big(-\frac{0.02\ln 10^2\sigma_i^2}{\alpha^2}\Big)z_u. \tag{3.11}$$

Therefore, the one and only unbiased estimate of $d_i^2$ is given by:

$$\begin{aligned}\hat{d}_i^2 &= \exp\Big(-\frac{2[\mathbf{Q}_\omega]_{ii}}{\alpha^2}\Big)\hat{d}^2_{\mathrm{ML}_i} \\ &= \exp\Big(-\frac{\ln(10)(P_i - P_0)}{5\alpha} - \frac{2[\mathbf{Q}_\omega]_{ii}}{\alpha^2}\Big).\end{aligned} \tag{3.12}$$

The corresponding covariance matrix is calculated using $E[(\hat{d}_i^2 - d_i^2)(\hat{d}_j^2 - d_j^2)]$. However, the covariance matrix in [42] could not work on a correlation case directly. Here, their work is adapted to a correlated case. Substituting (3.8) into (3.11), $\hat{d}_i^2$ can also be expressed as

$$\hat{d}_i{}^2 = d_i^2\exp(-\frac{2n_{r_i}}{\alpha} - \frac{2[\mathbf{Q}_\omega]_{ii}}{\alpha^2}). \tag{3.13}$$

Then the covariance matrix for $\hat{d}_i{}^2$ denoted by $\mathbf{Q}_r$ is given by

$$
\begin{aligned}
[\mathbf{Q}_r]_{ij} &= \mathbb{E}[(\hat{d}_i^2 - d_i^2)(\hat{d}_j^2 - d_j^2)] \\
&= \mathbb{E}\Big[d_i^2 d_j^2 \Big( \exp\Big( -\frac{2n_{r_i}}{\alpha} - \frac{2[\mathbf{Q}_r]_{ii}}{\alpha^2} \Big) - 1 \Big) \Big( \exp\Big( -\frac{2n_{r_j}}{\alpha} - \frac{2[\mathbf{Q}_r]_{jj}}{\alpha} \Big) - 1 \Big) \Big] \\
&= d_i^2 d_j^2 \Big\{ \exp\Big( -\frac{2([\mathbf{Q}_r]_{ii} + [\mathbf{Q}_r]_{jj})}{\alpha^2} \Big) \mathbb{E}\Big[ \exp\Big( -\frac{2(n_{r_i} + n_{r_j})}{\alpha} \Big) \Big] \\
&\quad - \exp\Big( -\frac{2[\mathbf{Q}_r]_{jj}}{\alpha^2} \Big) E\Big[ \exp\Big( -\frac{2n_{r_j}}{\alpha} \Big) \Big] - \exp\Big( -\frac{2[\mathbf{Q}_r]_{ii}}{\alpha^2} \Big) E\Big[ \exp\Big( -\frac{2n_{r_i}}{\alpha} \Big) \Big] + 1 \Big\}.
\end{aligned} \tag{3.14}
$$

In particular,

$$
\begin{aligned}
\mathbb{E}\Big[ \exp\Big( -\frac{2n_{r_i}}{\alpha} \Big) \Big] &= \frac{1}{\sqrt{2\pi\sigma}} \int \exp\Big( -\frac{2n_{r_i}}{\alpha} \Big) \exp\Big( -\frac{n_{r_i}^2}{0.02(\ln(10))^2 \sigma^2} \Big) \mathrm{d}n_{r_i} \\
&= \exp\Big( \frac{0.02(\ln(10))^2 \sigma_i^2}{\alpha^2} \Big) \frac{1}{\sqrt{2\pi\sigma}} \int \exp\Big( -\frac{\big(n_{r_i} + \frac{0.02(\ln(10))^2 \sigma_i^2}{\alpha}\big)^2}{0.02(\ln(10))^2 \sigma_i^2} \Big) \mathrm{d}n_{r_i} \\
&= \exp\Big( \frac{0.02(\ln(10))^2 \sigma^2}{\alpha^2} \Big) \\
&= \exp\Big( -\frac{2[\mathbf{Q}_r]_{ii}}{\alpha^2} \Big).
\end{aligned} \tag{3.15}
$$

Therefore, (3.14) can be simplified as

$$
[\mathbf{Q}_r]_{ij} = d_i^2 d_j^2 \{ \exp( -\frac{2([\mathbf{Q}_r]_{ii} + [\mathbf{Q}_r]_{jj})}{\alpha^2} ) \mathbb{E}[ \exp( -\frac{2(n_{r_i} + n_{r_j})}{\alpha} )] - 1 \}. \tag{3.16}
$$

If $i = j$, $[\mathbf{Q}_r]_{ii}$ is the variance of the distance estimate, and (3.15) can be written as

$$
[\mathbf{Q}_r]_{ii} = d_i^4 ( \exp( \frac{0.04 \ln^2(10) \sigma_i^2}{\alpha^2} ) - 1 ). \tag{3.17}
$$

If $i \neq j$, $(n_{r_i} + n_{r_j})$ is still gaussian, thus, similar to (3.15)

$$
\mathbb{E}\Big[ \exp\Big( -\frac{2(n_{r_i} + n_{r_j})}{\alpha} \Big) \Big] = \exp\Big( \frac{2}{\alpha^2} \mathrm{Var}(n_{r_i} + n_{r_j}) \Big). \tag{3.18}
$$

The variance of $(n_{r_i} + n_{r_j})$ can be expressed as

$$\text{Var}(n_{r_i} + n_{r_j}) = \text{Var}(n_{r_i}) + \text{Var}(n_{r_j}) + 2\text{Cov}(n_{r_i}, n_{r_j}). \tag{3.19}$$

Thus, when $i \neq j$,

$$\mathbb{E}[\exp(-\frac{2(n_{r_i} + n_{r_j})}{\alpha})] = \exp\Big(\frac{2}{\alpha^2}([\mathbf{Q}_\omega]_{ii} + [\mathbf{Q}_\omega]_{jj} + 2[\mathbf{Q}_\omega]_{ij})\Big). \tag{3.20}$$

Substituting (3.20) into (3.14) gives

$$[\mathbf{Q}_r]_{ij} = \begin{cases} d_i^4\Big(\exp\Big(\frac{0.04(\ln 10)^2\sigma_i^2}{\alpha^2}\Big) - 1\Big) & \text{if } i = j \\ d_i^2 d_j^2\Big(\exp\Big(\frac{0.04(\ln 10)^2\rho_{ij}\sigma_i\sigma_j}{\alpha^2}\Big) - 1\Big) & \text{if } i \neq j. \end{cases} \tag{3.21}$$

In practice, as the true $d_i$ is not available, further approximation is necessary in order to make the problem solvable. So far, the ML estimate and the unbiased estimate is available. And the unbiased estimate is better than the ML estimate in the sense of better mean and variance. Thus, $\hat{d}_i^2$ is used to approximate the true $d_i^2$. When the target is far away from the node, the variance of the unbiased estimate tends to increase, thus the covariance matrix is less accurate. The simulation in Section 3.5 shows the CWLS method with the approximate covariance matrix can reach the Cramer Rao Lower Bound (CRLB).

With the unbiased estimate of $d_i^2$ available, the distance based LS is formulated the same way shown in Section 2.2.1. In an ideal situation, when the distance is free of noise, the circles in Fig.2.1 should have a single common intersection. Expanding $d_i^2 = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ and reformatting it into matrix gives

$$\mathbf{A}\boldsymbol{\vartheta} = \mathbf{b} \tag{3.22}$$

where

$$\mathbf{A} = \begin{bmatrix} -2x_1 & -2y_1 & 1 \\ \vdots & \vdots & \vdots \\ -2x_N & -2y_N & 1 \end{bmatrix},$$

$$\boldsymbol{\vartheta} = \begin{bmatrix} x \\ y \\ x^2 + y^2 \end{bmatrix}^T$$

and

$$\mathbf{b} = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 \\ \vdots \\ d_N^2 - x_N^2 - y_N^2 \end{bmatrix}.$$

Note that $\mathbf{A}$ is a matrix with nodes' location, $\boldsymbol{\vartheta}$ is a vector containing the parameter to be estimated and $\mathbf{b}$ is the observation vector contains the distance. Let

$$\hat{\mathbf{b}} = \mathbf{b} + \Delta\mathbf{b}, \tag{3.23}$$

where $\hat{\mathbf{b}}_i = \hat{d}_i^2 - x_i^2 - y_i^2$, and $\Delta\mathbf{b}_i = \hat{d}_i^2 - d_i^2$ is the error vector in the unbiased distance estimate. The covariance of $\Delta\mathbf{b}$ is $\mathbf{Q}_r$. With the unbiased distance square estimate available, the LS solution can be obtained by minimizing

$$J(\boldsymbol{\vartheta}) = (\mathbf{A}\boldsymbol{\vartheta} - \hat{\mathbf{b}})^T(\mathbf{A}\boldsymbol{\vartheta} - \hat{\mathbf{b}}) \tag{3.24}$$

The solution to (3.24) is simply

$$\hat{\boldsymbol{\vartheta}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\hat{\mathbf{b}}. \tag{3.25}$$

Although (3.22) is a linear equation in $\boldsymbol{\vartheta}$, the third component of $\boldsymbol{\vartheta}$ is in fact related to the target location $\mathbf{u}$. Taking this fact into consideration is necessary. As mentioned in Section 2.2.1, in the literature, it has been used to formulate a LS with a Lagrange function. Here, inspired by the method used in [67], the CWLS solution can be obtained by

$$\begin{aligned} &\text{minimize} \quad J(\boldsymbol{\vartheta}) = (\mathbf{A}\boldsymbol{\vartheta} - \mathbf{b})^T\mathbf{Q}_r^{-1}(\mathbf{A}\boldsymbol{\vartheta} - \mathbf{b}) \\ &\text{subject to} \quad \vartheta_3 = \vartheta_1^2 + \vartheta_2^2. \end{aligned} \tag{3.26}$$

Substituting the constraint into the objective function, replacing $(\boldsymbol{\vartheta}_1^2 + \boldsymbol{\vartheta}_2^2)$ by $\mathbf{u}^T\mathbf{u}$, and partitioning $\mathbf{A}$ as $[\overline{\mathbf{A}}, \mathbf{1}_N]$, where $\overline{\mathbf{A}}$ denotes the first two columns of $\mathbf{A}$, and $\mathbf{1}_N$ is an all one vector. Then, the cost function of (3.26) can be rewritten as

$$J(\mathbf{u}) = \boldsymbol{\gamma}_r^T\mathbf{Q}_r^{-1}\boldsymbol{\gamma}_r. \tag{3.27}$$

where $\boldsymbol{\gamma}_r = \overline{\mathbf{A}}\mathbf{u} + \mathbf{1}_N(\mathbf{u}^T\mathbf{u}) - \mathbf{b}$. Now, (3.27) has a similar form as (2.12) as shown in Section 2.2.1. To find a solution to minimize $J(\mathbf{u})$ with respect to $\mathbf{u}$, numerical algorithms may be adopted to solve these nonlinear equations. Since an initial solution can be obtained by using ordinary LS (3.25), Newton's method is chosen to give a solution. Compared with other iterative methods, i.e., gradient method, Newton's method converges much faster. If the minimized function has a continuous second derivative, then Newtons method has order-two convergence [68].

The second-order Taylor series expansion of $J(\mathbf{u})$ is [69]

$$J(\mathbf{u}) \approx J(\mathbf{u}_o) + (\mathbf{u} - \mathbf{u}_o)^T\frac{\partial J(\mathbf{u}_o)}{\partial \mathbf{u}} + \frac{1}{2}(\mathbf{u} - \mathbf{u}_o)^T\frac{\partial^2 J(\mathbf{u}_o)}{\partial \mathbf{u}\partial \mathbf{u}^T})(\mathbf{u} - \mathbf{u}_o). \tag{3.28}$$

A necessary condition for the minimum of $J(\mathbf{u})$ is that $\partial J(\mathbf{u})/\partial \mathbf{u} = \mathbf{0}$. Denote $\boldsymbol{\Psi} = \partial^2 J(\mathbf{u}_o)/\partial \mathbf{u}\partial \mathbf{u}^T$ and $\boldsymbol{\eta} = \partial J(\mathbf{u}_o)/\partial \mathbf{u}$, from (3.28), it follows that

$$\frac{\partial J(\mathbf{u})}{\partial \mathbf{u}} = \boldsymbol{\eta} + \boldsymbol{\Psi}(\mathbf{u} - \mathbf{u}_o) = 0. \tag{3.29}$$

Starting with initial guess $\mathbf{u}(0)$, Newtons method proceeds by the iteration

$$\mathbf{u}(t+1) = \mathbf{u}(t) - \boldsymbol{\Psi}^{-1}\boldsymbol{\eta}. \tag{3.30}$$

where $t$ is the iteration step. The first order derivative of (3.27) with respect to $\mathbf{u}$ is

$$\begin{aligned} \frac{\partial J(\mathbf{u})}{\partial \mathbf{u}} &= \frac{\partial \boldsymbol{\gamma}_r^T}{\partial \mathbf{u}} \mathbf{Q}_r^{-1} \boldsymbol{\gamma}_r + \boldsymbol{\gamma}_r^T \frac{\partial (\mathbf{Q_r}^{-1} \boldsymbol{\gamma}_r)}{\partial \mathbf{u}} \\ &= 2\boldsymbol{\gamma}_r^T \mathbf{Q}_r^{-1} \boldsymbol{\Gamma}_r. \end{aligned} \tag{3.31}$$

where $\boldsymbol{\Gamma}_r = \frac{\partial \boldsymbol{\gamma}_r}{\partial \mathbf{u}} = \overline{\mathbf{A}} + 2(\mathbf{u}\mathbf{1}_N^T)^T$.

The second order derivative of (3.27) with respect to $\mathbf{u}$ is

$$\frac{\partial}{\partial \mathbf{u}^T}\left(\frac{\partial J(\mathbf{u})}{\partial \mathbf{u}}\right) = 2\boldsymbol{\Gamma}_r \mathbf{Q_r}^{-1} \boldsymbol{\Gamma}_r + 4\boldsymbol{\gamma}_r^T \mathbf{Q_r}^{-1} \mathbf{1}_N \mathbf{I}_2. \tag{3.32}$$

where $\mathbf{I}_2$ is an identity matrix of dimension two.

Using (3.31) and (3.32), $\boldsymbol{\eta}$ and $\boldsymbol{\Psi}$ are given as

$$\boldsymbol{\eta} = \frac{\partial J(\mathbf{u})}{\partial \mathbf{u}} \mid_{\mathbf{u}=\mathbf{u}_o}, \tag{3.33}$$

$$\boldsymbol{\Psi} = \frac{\partial^2 J(\mathbf{u})}{\partial \mathbf{u} \partial \mathbf{u}^T} \mid_{\mathbf{u}=\mathbf{u}_o}. \tag{3.34}$$

A critical problem with Newtons method is that it cannot guarantee the convergence of the iteration. However, in the localization problem, the convergence is easy to detect [40]. Substitute the estimate $\hat{\mathbf{x}}$ at current iteration into the ML cost function (2.11) to find the value of the cost function, $f_t$. If $|f_t - f_{t-1}|/f_{t-1} < \iota$ ($\iota$ is a given small positive number, say, 0.001), then stop. The standard LS solution (3.6) is chosen as the initial estimate of $\mathbf{u}$. In the simulation, a maximum iteration number is set as the stop criterion.

The procedures of CWLS is summarized in Algorithm 3.3.1.

**Algorithm 3.3.1:** CWLS for RSS based localization

Setup Problem:

• Initialize location of each node.
• Collect the RSS measurements.
• Compute the unbiased distance estimate and the corresponding covariance.
• Use the LS (3.25) as the initial estimate $\mathbf{u}_o$.

**for** *each iteration* **do**
    Update $\mathbf{u}$ using (3.31) and (3.32).;
    **if** *maximum iteration is reached* **then**
        Terminate iteration;
    **end**
**end**

### 3.3.2 Unknown Transmit Power

In the case of unknown transmit power, we select a reference node and employ the DRSS measurements [54], [56].

As introduced in Section 2.2.1, the DRSS measurement is expressed as:

$$q_{i1} = P_i - P_1 = -10\alpha \log_{10} r_{i1} + n_{v_{i1}}, \quad i = 2, \ldots, N. \tag{3.35}$$

where $r_{i1} = d_i/d_1$, and $n_{v_{i1}} = n_i - n_1$ is the zero mean Gaussian random variable.

Specifically, the covariance matrix $\mathbf{Q}_v$ is correlated, given by

$$\mathbf{Q}_v = \mathbf{\Theta}\mathbf{Q}\mathbf{\Theta}^T, \tag{3.36}$$

where $\mathbf{\Theta}$ is a $(N-1) \times N$ matrix

$$\mathbf{\Theta} = \begin{bmatrix} -1 & 1 & 0 & \ldots & 0 \\ -1 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & \ldots & 1 \end{bmatrix}.$$

Similar to (3.3), the observed differential path loss can be written as

$$0.1\ln(10)(P_i - P_1) = -\alpha \ln r_{i1} + n_{e_{i1}}, \tag{3.37}$$

where $n_{e_{i1}} = 0.1(\ln 10)n_{v_{i1}}$, and its covariance matrix is

$$\mathbf{Q}_e = 0.01(\ln 10)^2 \mathbf{Q}_v. \tag{3.38}$$

Compared with the RSS measurement, the DRSS measurement model have the same form of the RSS measurement with zero value transmit power. Thus, the unbiased estimate of $r_{i1}^2$ can be obtained by using the results in [42], and the corresponding covariance is obtained using the adapted correlated covariance described in Section 3.3.1.

Applying the same approach in (3.12) [42], the unbiased estimator of $r_{i1}^2$ is given by

$$\hat{r}_{i1}^2 = \exp\Big( - \frac{0.2\ln(10)(P_i - P_1)}{\alpha} - \frac{2[\mathbf{Q}_e]_{ii}}{\alpha^2} \Big). \tag{3.39}$$

Similar to the process from (3.15) to (3.21), the covariance matrix for $\hat{r}_{i1}^2$ is

$$[\mathbf{Q}_d]_{ij} = \begin{cases} r_{i1}^4 (\exp(\frac{4[\mathbf{Q}_e]_{ii}}{\alpha^2}) - 1) & \text{if } i = j \\ r_{i1}^2 r_{j1}^2 (\exp(\frac{4[\mathbf{Q}_e]_{ij}}{\alpha^2}) - 1) & \text{if } i \neq j \end{cases}. \tag{3.40}$$

Now, the weighted least squares (WLS) solution is formulated similar to the LS shown in Section 2.2.1. From the distance ratio, we have $r_{i1}^2 d_1^2 = d_i^2$. Expanding $d_1$ and $d_i$ using (2.5) gives

$$(r_{i1}^2 - 1)d_1^2 = x_i^2 + y_i^2 - x_1^2 - y_1^2 - 2(x_i - x_1)x - 2(y_i - y_1)y. \tag{3.41}$$

Denote the error in $r_{i1}^2$ as $\Delta r_{i1}$. Substituting $r_{i1}^2 = \hat{r}_{i1}^2 - \Delta r_{i1}$ into (3.27) gives

$$d_1^2 \Delta r_{i1} = x_1^2 + y_1^2 - x_i^2 - y_i^2 + 2(x_i - x_1)x + 2(y_i - y_1)y + (1 - \hat{r}_{i1}^2)d_1^2. \tag{3.42}$$

Reorganising (3.42) and formatting it in matrix form gives [70]

$$d_1^2 \Delta \mathbf{r} = \mathbf{h} - \hat{\mathbf{G}} \boldsymbol{\varkappa}, \tag{3.43}$$

where

$$\hat{\mathbf{G}} = \begin{bmatrix} 2(x_1 - x_2) & 2(y_1 - y_2) & 1 - \hat{r}_{21}^2 \\ \vdots & \vdots & \vdots \\ 2(x_1 - x_N) & 2(y_1 - y_N) & 1 - \hat{r}_{N1}^2 \end{bmatrix},$$

$$\boldsymbol{\varkappa} = \begin{bmatrix} x \\ y \\ d_1^2 \end{bmatrix}$$

and

$$\mathbf{h} = \begin{bmatrix} x_1^2 + y_1^2 - x_2^2 - y_2^2 \\ \vdots \\ x_1^2 + y_1^2 - x_N^2 - y_N^2 \end{bmatrix}$$

and $\Delta \mathbf{r} = [\Delta r_{21}, \ldots, \Delta r_{N1}]^T$ is the error given by vector of $r_{i1}^2$. The covariance of $d_1^2 \Delta r_{i1}$ is $d_1^4 \mathbf{Q}_d$. Nonetheless, $d_1$ is unavailable. Since scaling the weighting matrix of the WLS will not influence the performance, $d_i^4$ is discarded.

As can be observed in $\boldsymbol{\varkappa}$, the intermediate value $\varkappa_3 = d_1^2$ is related to $\mathbf{u}$ and the reference node $\mathbf{c}_1$. The CWLS solution can be obtained as

$$\begin{aligned} &\text{minimize} J(\boldsymbol{\varkappa}) \quad = (\mathbf{G}\boldsymbol{\varkappa} - \mathbf{h})^T \mathbf{Q}_d^{-1} (\mathbf{G}\boldsymbol{\varkappa} - \mathbf{h}) \\ &\text{subject to} \quad \varkappa_3 = (\mathbf{u} - \mathbf{c}_1)^T (\mathbf{u} - \mathbf{c}_1). \end{aligned} \tag{3.44}$$

Partitioning $\hat{\mathbf{G}}$ as $[\overline{\mathbf{G}}_{12}, \overline{\mathbf{G}}_3]$, where $\overline{\mathbf{G}}_{12}$ denotes the first two columns of $\hat{\mathbf{G}}$ and $\overline{\mathbf{G}}_3$ represents

the third columns of $\hat{\mathbf{G}}$. Therefore, (3.44) can be rewritten as

$$J(\mathbf{u}) = \boldsymbol{\gamma}_d^T \mathbf{Q}_d^{-1} \boldsymbol{\gamma}_d. \tag{3.45}$$

where $\boldsymbol{\gamma}_d = \overline{\mathbf{G}}_{12}\mathbf{u} + \overline{\mathbf{G}}_3(\mathbf{u}^T\mathbf{u} + \mathbf{c}_1^T\mathbf{c}_1 - 2\mathbf{c}_1^T\mathbf{u}) - \mathbf{h}$.

Then the gradient and hessian matrix of $J(\mathbf{u})$ can be expressed as

$$\frac{\partial J(\mathbf{u})}{\partial \mathbf{u}} = 2\boldsymbol{\gamma}_d^T \mathbf{Q}_d^{-1} \boldsymbol{\Gamma}_d, \tag{3.46}$$

$$\frac{\partial^2 J(\mathbf{u})}{\partial \mathbf{u}^T \partial \mathbf{u}} = 2\boldsymbol{\Gamma}_d^T \mathbf{Q}_d^{-1} \boldsymbol{\Gamma}_d + 4\boldsymbol{\gamma}_2^T \mathbf{Q}_d^{-1} \mathbf{1}_{N-1} \mathbf{I}_2. \tag{3.47}$$

where $\boldsymbol{\Gamma}_d = \frac{\partial \boldsymbol{\gamma}_d}{\partial \mathbf{u}} = \overline{\mathbf{G}}_{12} + 2\overline{\mathbf{G}}_3\mathbf{u}^T - 2\overline{\mathbf{G}}_3\mathbf{c}_1^T$.

The procedures of the proposed CWLS for the DRSS based are summarized in Algorithm 3.3.2.

**Algorithm 3.3.2:** CWLS for DRSS based localization

Setup Problem:
- Initialize location of each node.
- Collect the DRSS measurements.
- Compute the unbiased estimate $\hat{r}_{i1}$ and the corresponding covariance.
- Use the first two elements of $(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{h}$ as the initial estimate $\mathbf{u}_o$.

**for** *each iteration* **do**
    Update $\mathbf{u}$ using (3.46) and (3.47).;
    **if** *maximum iteration is reached* **then**
        Stop iteration;
    **end**
**end**

In this section, the original work for the unbiased distance estimate in [42] is adapted to a correlated case. Further, based on the unbiased estimate, the CWLS methods for both RSS based and DRSS based localization are derived.

## 3.4 Complexity Analysis

The computational complexity of the proposed CWLS method based on the required number of flops is derived in this section. Assume that simple operations such as addition, subtraction,

| Algorithm | Complexity | Time(ms) |
|---|---|---|
| CWLS | $O(K_{\text{GN}}(3N^2+9N)+9N)$ | 0.69 |
| SDP-RSS | $O(K_{\text{SDP}}N^4\log(1/\varepsilon)$ | 353 |
| LLS | $O(6N^2+12N)$ | 0.16 |
| UT-GTRS | $O(36K_{\text{GTR}}+34N)$ | 1.17 |

Table 3.1: Complexity and average running time of the algorithms.

multiplication and division cost one flop, and the results are simplified by ignoring low order terms.

We compare the complexity of the LLS [16], SDP-RSS-I [18], Unscented transform-Generalized Trust Region Subproblem (UT-GTRS) [47] and the proposed CWLS in this section. The complexity of the SDP-RSS-I can be computed as $O(K_{\text{SDP}}N^4\log(1/\varepsilon))$, where $\varepsilon$ is the accuracy tolerance [71] and $K_{\text{SDP}}$ is the iteration number of the SDP method. In the LLS, the estimate is computed using $\hat{\boldsymbol{\vartheta}}=(\mathbf{A}^T\mathbf{Q}_r^{-1}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Q}_r^{-1}\hat{\mathbf{b}}$. This includes five matrix multiplications and one inverse of matrix. If $\mathbf{A}\in\mathbb{R}^{m\times n}$, $\mathbf{b}\in\mathbb{R}^{m\times 1}$ and $\mathbf{Q}_r\in\mathbb{R}^{m\times m}$, then the computational complexity is $O(2m^2n+mn^2+mn+n^3+n^2)$ [72]. Let $m=N$ and $n=3$, omitting the constant, the complexity for LLS is $O(6N^2+12N)$. For the UT-GTRS method, suppose that the bisection search takes $K_{\text{GTR}}$ steps, then the total sum of the proposed approach can be approximated as $O(36K_{\text{GTR}}+34N)$ [47]. For the proposed approach, the complexity for initialisation (3.4) is $O(9N)$ and the complexity for iteration is $O(K_{\text{GN}}(3N^2+9N))$ where $K_{\text{GN}}$ is the iteration number of the CWLS. Table 3.1 summarises the complexity of the different approaches.

The average running time of different approaches for the same scenario considered in Section 3.5, Fig.3.1 are also measured. The algorithms are implemented in Matlab 2012 using an Intel i5-430m processor. The average running time of algorithms for 100 Monte Carlo simulations is shown in Table 3.1. The SDP method is programmed using the tool box CVX [45]. The programming of other algorithms follows exactly the same steps as stated in their papers. The proposed algorithm has a reasonable complexity compared to other approaches.

## 3.5 Numerical results

Monte Carlo simulations are performed to evaluate the performance of the CWLS algorithm by comparing with other existing algorithms, as well as the CRLB. A 20m by 20m area is considered

for the simulations. Eight nodes are placed at fixed positions $\mathbf{c}_1 = (0,0)^T$, $\mathbf{c}_2 = (0,10)^T$, $\mathbf{c}_3 = (0,20)^T$, $\mathbf{c}_4 = (10,20)^T$, $\mathbf{c}_5 = (20,20)^T$, $\mathbf{c}_6 = (10,0)^T$, $\mathbf{c}_7 = (20,10)^T$, and $\mathbf{c}_8 = (20,0)^T$ as shown in Fig.3.1. In the simulations for every realisation, the transmit power and the PLE are -10dBm and 4, respectively, unless stated otherwise. The target is located at (6,14)m, which is inside the square bounded by the sensor coordinates. It is assumed that all the measurements have the same variance, $\sigma_i = \sigma$ and the same correlation coefficient $\rho_{ij} = \rho$. All simulations included 5,000 independent runs.

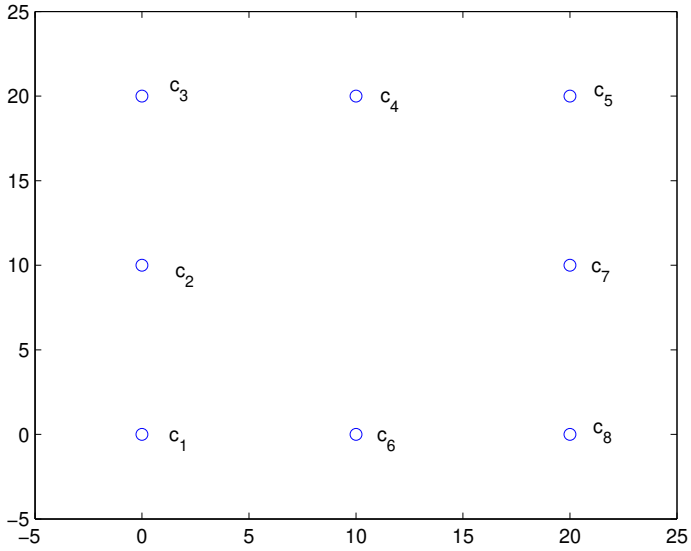



Figure 3.1: Nodes topology.

### 3.5.1 Known Transmit Power

In this section, the proposed CWLS algorithm is compared with the CRLB, the LLS, SDP-RSS-I, SDP-RSS-II [46] and UT-GTRS when the transmit power is known as *a priori*.

Fig.3.2 shows the root-mean-square-root (RMSE) of the location estimate for different approaches versus the variances of the shadow noise when RSS measurements are uncorrelated ($\rho = 0$). The PLE in this simulation is 4. The RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{1}{M_{\text{run}}}||\mathbf{u} - \hat{\mathbf{u}}||}, \tag{3.48}$$

where $M_{\text{run}}$ is the number of Monte Carlo trials and $||\cdot||$ represents the norm operator. The

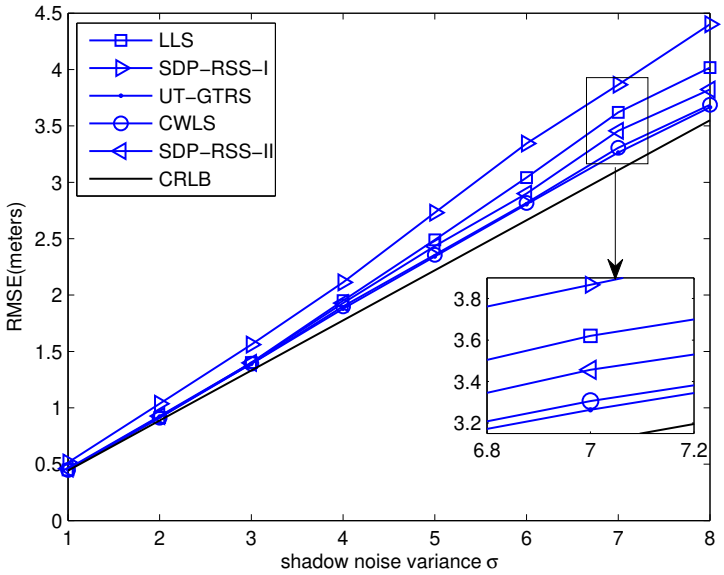



Figure 3.2: RMSE of the estimators versus uncorrelated shadow noise using RSS measures.

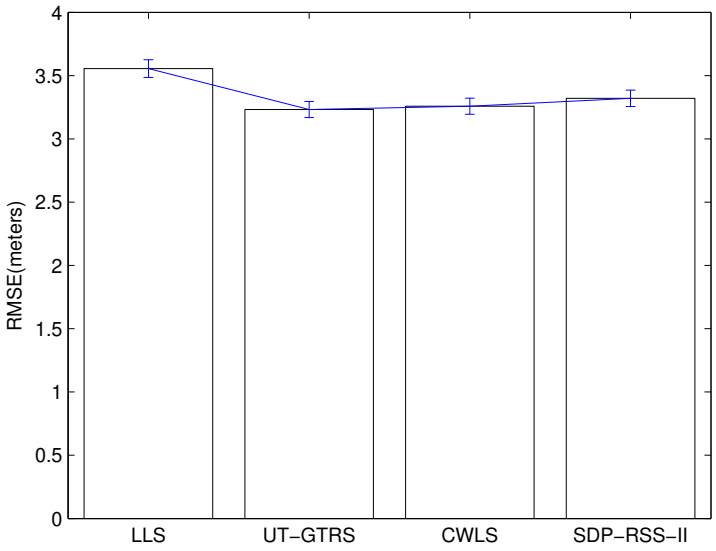



Figure 3.3: RMSE of the estimators at $\sigma = 7$ with error bar.

| Algorithm | p-value for x-axis | p-value for y-axis | $H_0$ for x-axis | $H_0$ for y-axis |
|---|---|---|---|---|
| SDP-RSS-II/CWLS | $0.008 \times 10^{-9}$ | $0.108 \times 10^{-9}$ | 1 | 1 |
| LLS/CWLS | 0 | $0.175 \times 10^{-5}$ | 1 | 1 |
| UT-GTRS/CWLS | 0.91 | 0.89 | 0 | 0 |

Table 3.2: Results for two sample t-test between CWLS and other algorithms.

| Algorithm | CWLS | LLS | UT-GTRS | SDP-RSS-II |
|---|---|---|---|---|
| NRMSE for x-axis at position $\mathbf{a}_1$ | 0.47546 | 0.49624 | 0.47108 | 0.47652 |
| NRMSE for y-axis at position $\mathbf{a}_1$ | 0.15606 | 0.15713 | 0.15491 | 0.15677 |
| NRMSE for x-axis at position $\mathbf{a}_2$ | 0.23840 | 0.28941 | 0.23670 | 0.24939 |
| NRMSE for y-axis at position $\mathbf{a}_2$ | 0.23759 | 0.28451 | 0.23589 | 0.24928 |
| NRMSE for x-axis at position $\mathbf{a}_3$ | 0.47803 | 0.48625 | 0.47225 | 0.52537 |
| NRMSE for y-axis at position $\mathbf{a}_3$ | 0.20195 | 0.26698 | 0.19716 | 0.21025 |
| NRMSE for x-axis at position $\mathbf{a}_4$ | 0.33360 | 0.40274 | 0.33060 | 0.34623 |
| NRMSE for y-axis at position $\mathbf{a}_4$ | 0.19291 | 0.21853 | 0.19085 | 0.19967 |

Table 3.3: Normalised RMSE (NRMSE) of different methods at $\sigma = 7$.

RMSE with error bar for these methods when $\sigma = 7$dB is shown in Fig.3.3. The error bar gives a general idea of how precise these data are. The simulation results show that the accuracy of the CWLS solution can approach the CRLB for the whole range of $\sigma$. It is also seen clearly that the proposed CWLS method outperforms LLS, SDP-RSS-I and SDP-RSS-II. The SDP-RSS-I algorithm has the worst performance because extra noise is introduced into the signal model twice during the semidefinite relaxation. As $\sigma$ grows, the performance of the LLS decreases faster than that of the CRLB. The reason for this is that, in the LLS, the second LS is biased when $\sigma$ is sufficiently large [50]. Using a two sample t-test, we test the hypothesis $H_0$: the location estimate of two algorithms come from distributions with equal means. $H_0 = 0$ indicates that the null hypothesis ("means are equal") cannot be rejected at the 5% level. The CWLS is compared with other algorithms using this method. The results are summarized in Table 3.2. The results show that the CWLS method have the same mean with the GTRS method at a level of significance of 5%. The CWLS has similar performance to the UT-GTRS, however, it consumes only half of the running time compared with the UT-GTRS.

In the next simulation, the RMSE is normalised to show how well the CWLS method works

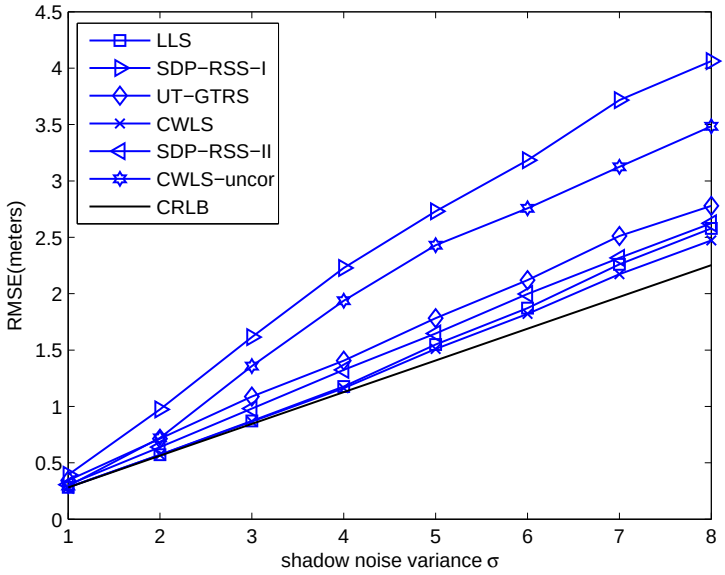



Figure 3.4: RMSE of the estimators versus correlated shadow noise using RSS measures.

compared with other methods. The target is located in different places in this simulation. Since the topology is symmetric, four positions are chosen where $\mathbf{a}_1 = (5, 15)$, $\mathbf{a}_2 = (10, 10)$, $\mathbf{a}_3 = (4, 10)$ and $\mathbf{a}_4 = (7, 12)$. The results are recorded in Table 3.3. The results show that the proposed CWLS have better performance than the LLS and SDP methods.

In Fig.3.4, the same experiment in Fig.3.2 is repeated, except that shadowing noise is correlated ($\rho = 0.6$). The CWLS-uncor in Fig.3.4 is the CWLS estimator using a diagonal covariance matrix ($\rho = 0$). Compare the CRLB between Fig.3.2 and Fig.3.4, the localization accuracy improves significantly. Among all the algorithms, SDP-RSS-I and UT-GTRS do not consider the correlation among RSS measurements or cannot be easily adapted to do so. Thus, they do not have much improvement. As shown in the figure, when the correlation is taken iinto consideration, the proposed CWLS outperforms other methods.

However, the assumption that the correlation coefficient is available to the system is not always valid in practice. Here, the performance of the CWLS is examined with respect to the correlation coefficient $\rho$. Assume that only an inaccurate estimate $\hat{\rho}$ is available to system. It is modelled as the true $\rho$ added with a random variable

$$\hat{\rho} = \rho + \Delta\rho, \tag{3.49}$$

where $\Delta\rho$ is a zero-mean Gaussian random variable with variance of 0.1. If $\hat{\rho}$ exceeds [0.2 0.8], it is adjusted to 0.2 or 0.8. The same simulation parameters as in Fig.3.2 are considered. The shadowing noise $\sigma$ is 5dB in the simulation. The RMSE of the CWLS with inaccurate $\rho$ is 1.7m, while the RMSE of the CWLS with true $\rho$ is 1.51m. The performance of the CWLS is still acceptable compared with CWLS-uncor with an RMSE of 2.43m.

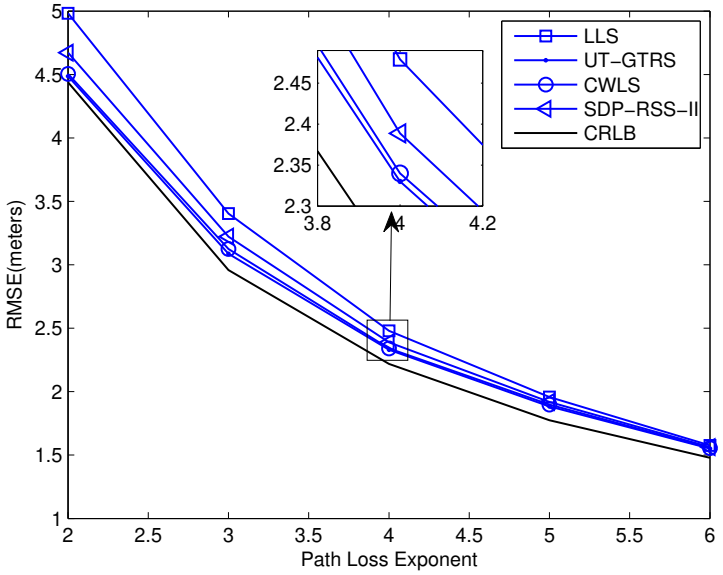



Figure 3.5: RMSE of the estimators versus the PLE using RSS measures when $\rho = 0$.

In Fig.3.5, the RMSEs for different PLE from 2 to 6 at 5dB shadow noise are plotted. Other experimental parameters are the same as in the first test. Again, it can be observed that the CWLS is very close to the CRLB. The localization error decreases as PLE increases, which is indicated by the CRLB. The RMSE of LLS and SDP-RSS attains the CRLB when the path loss exponent is greater than or equal to 4.

Fig.3.6 compares the RMSEs of different algorithms when the number of anchor nodes varies from 4 to 8. The RMSE with error bar for these methods when 5 nodes ($\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$, $\mathbf{c}_4$, $\mathbf{c}_5$) are used is shown in Fig.3.3. The node is deleted from node $\mathbf{c}_8$ to $\mathbf{c}_4$. $\rho$ is assumed to be zero in this simulation. It can be seen that the performance of all algorithms improves as the number of nodes increases. The CWLS algorithm is worse than the GTRS method but better than the rest. Note that the performance of SDP, GTRS and CWLS outperform the CRLB. The simulation in [52] shows that the GTRS method is biased. As for SDP, simulations in [46] also shows that it is biased. In Appendix E, it is proved that the CWLS is unbiased only when the variance is small. In Table 3.4, the RMSE of CWLS and GTRS is recorded at different noise level when five

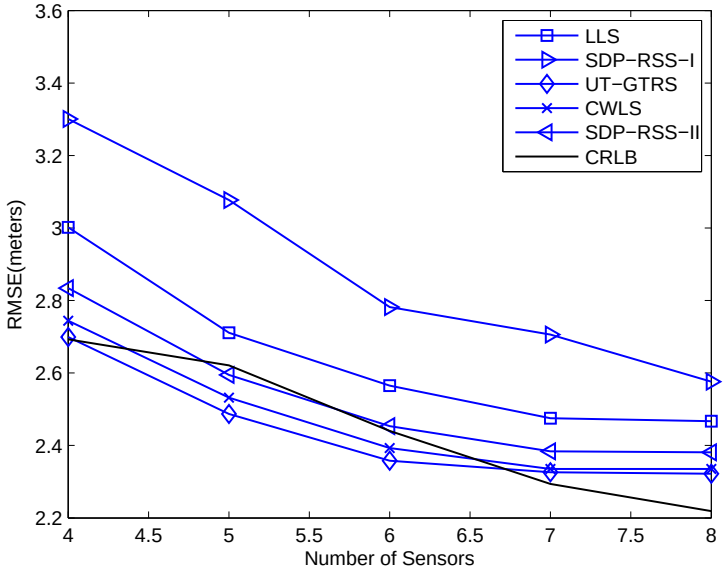



Figure 3.6: RMSE of the estimators versus number of nodes with shadowing noise $\sigma = 5$, PLE $\alpha = 4$ and $\rho = 0$ using RSS measures.

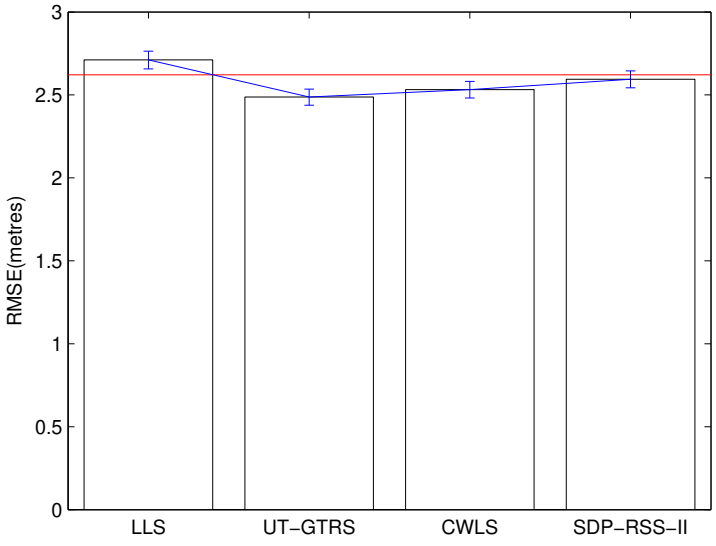



Figure 3.7: RMSE of the estimators when 5 nodes are used with error bar. Red line in the figure represents the CRLB.

| Algorithm | $\sigma = 0.001$ | $\sigma = 0.1$ | $\sigma = 1$ | $\sigma = 3$ | $\sigma = 5$ |
|---|---|---|---|---|---|
| UT-GTRS | $10.83 \times 10^{-4}$ | $5.776 \times 10^{-2}$ | 0.546 | 1.532 | 2.457 |
| CWLS | $5.239 \times 10^{-4}$ | $5.254 \times 10^{-2}$ | 0.523 | 1.521 | 2.502 |
| CRLB | $5.241 \times 10^{-4}$ | $5.242 \times 10^{-2}$ | 0.524 | 1.572 | 2.621 |

Table 3.4: Results for two sample t-test between CWLS and other algorithms.

nodes are deployed. Other parameter settings are the same as in Fig.3.5. As can be seen in Table 3.4, when $\sigma$ is quit small ($\sigma < 0.1$), the CWLS method is close to CRLB. As $\sigma$ increases, the gap between CWLS and CRLB increases. Thus, the CWLS is a biased estimator when shadowing noise is large enough. The phenomena that the performance of an algorithm is better than CRLB happens in other study [23]. Since the estimate is biased, the CRLB cannot be expected to provide a lower bound on the accuracy.

### 3.5.2 Unknown Transmit Power

In this section, we compare the proposed algorithm with the CRLB, LS [22] and SDP-URSS [23], when the transmit power is unknown. The node (0,0) is chosen as the reference node.

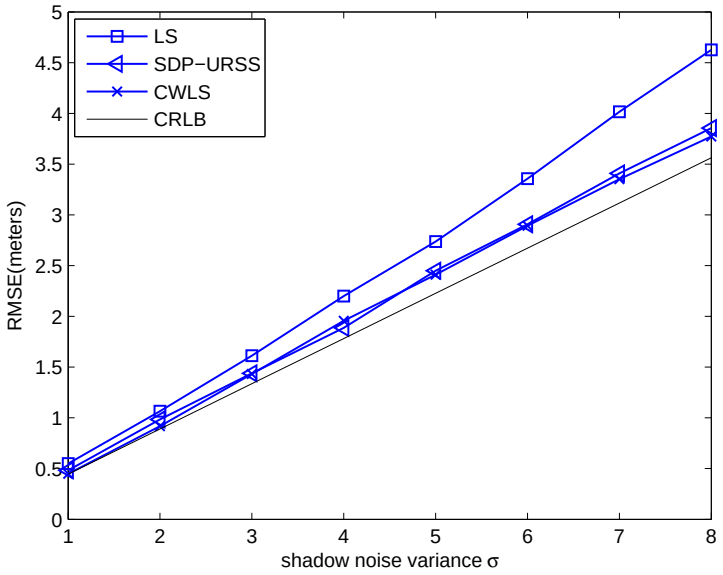



Figure 3.8: RMSE of the estimators versus uncorrelated shadow noise using DRSS measures.

Fig.3.8 shows the RMSEs of the location estimate for different approaches versus the shadow noise variance. The PLE is 4 in this simulation. The results show that the CWLS method has

better performance than the LS. SDP-URSS has similar performance to CWLS, but CWLS has a significant advantage in computational cost. At a relatively high shadowing variance, CWLS outperforms SDP-URSS slightly. The reason is that the Taylor series expansion is used during the relaxation where a biased noise is introduced. Also, the SDP estimator is biased.

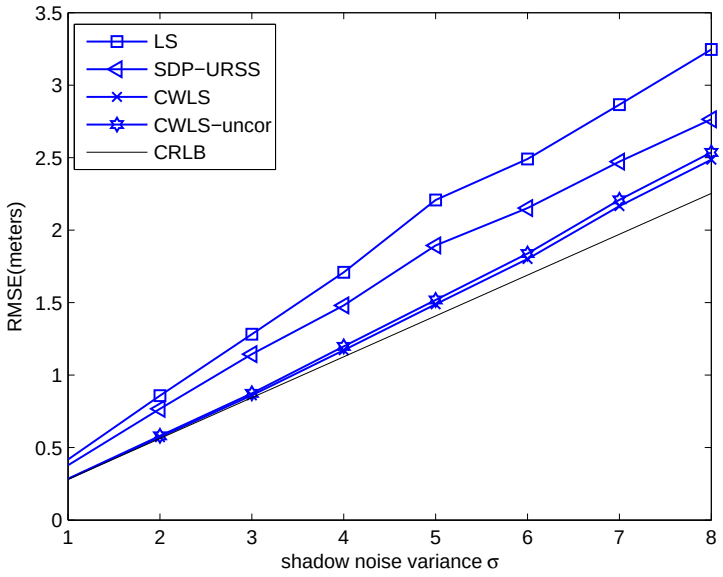



Figure 3.9: RMSE of the estimators versus correlated shadow noise using DRSS measures.

In Fig.3.9, the same experiment in Fig.3.4 is repeated, except shadowing noise is correlated ($\rho = 0.6$). The CWLS-uncor in Fig.3.9 is the CWLS estimator using uncorrelated RSS measurements. Similar to Fig.3.4, the localization accuracy increases when RSS measurements are correlated. One interesting point is that the performance of CWLS and CWLS-uncor is similar. This is because the noise variance matrix of the DRSS measurement is correlated with $\rho = 0.5$. Thus, when the DRSS measurement is employed in the localization problem, it is not necessary to require the knowledge of the correlation coefficient.

Fig.3.10 shows the RMSEs for different PLE from 2 to 6 when transmit power is unknown to the system. Similar to Fig.3.5, the CWLS outperforms the SDP while the path loss exponent is small. This advantage decreases as the PLE increases. The reason is that under the same variance assumption while PLE increase, the noise has smaller impact on the accuracy. This can be seen clearly from the path loss signal model.

Fig.3.11 compares the RMSEs of different algorithms when the number of anchors increases from 4 to 8. $\rho$ is assumed to be zero in this simulation, and shadowing noises $\sigma$ and PLE are 5dB and 4, respectively. It can be seen that the proposed algorithm always has better performance than other

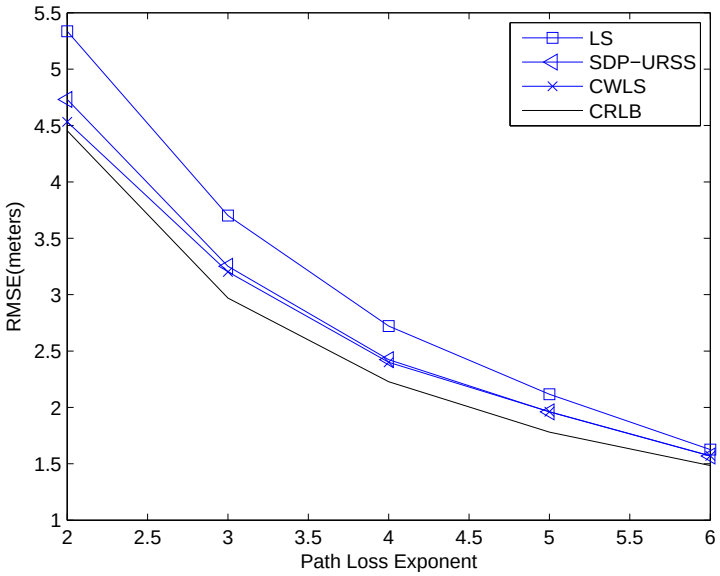



Figure 3.10: RMSE of the estimators versus the PLE with uncorrelated shadow noise $\sigma = 5$ using DRSS measures.

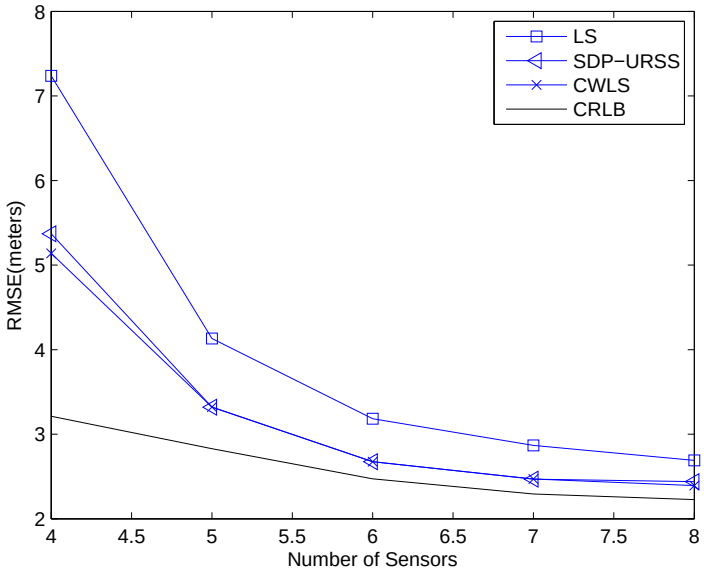



Figure 3.11: RMSE of the estimators versus number of nodes using DRSS measures.

algorithms. One interesting phenomenon is that, when only four anchors are deployed, the RMSE of all algorithms is very large. This is because there is little redundant information from the extra measurement that could be used to reduce the negative effect of the shadowing noise.

## 3.6 Summary

In this chapter, we have studied the target localization problem when the RSS measurements are assumed to be correlated. Firstly, the unbiased distance estimator [42] is extended to a correlated case. Then, an effective CWLS method inspired by the constrained total LS for TDOA based localization [67] is proposed to estimate the target position based on the distance estimation. Further, the CWLS is extend to solve the DRSS based localization problem. Simulation results have shown that the CWLS outperforms most existing algorithms in terms of higher localization accuracy. Under the correlated noise assumption, the CWLS has the best performance among all algorithms. Also, the CWLS is easy to implement with low computation cost. The accuracy of the CWLS achieves the CRLB in most scenarios.

# Chapter 4

# Distributed RSS-based Localization with Unknown Channel Parameters

## 4.1 Introduction

In this chapter, the target localization problem with received signal strength (RSS) measurements when the channel parameter, i.e. the path loss exponent (PLE), is unknown, is dealt with using particle swarm optimisation (PSO). As mentioned in Section 2.1, determining the PLE requires a large number of experiments, which is expensive or even impractical under certain circumstances. Most of the existing algorithms tackle RSS based localization problem with unknown PLE have been reviewed in Section 2.2.2. They all need prior information on the PLE for initialization. To avoid requiring initial on the PLE, PSO, a global optimisation algorithm, is employed to optimise the non-linear and non-convex ML cost function. The use of PSO in the RSS-based localization problem with unknown PLE has been studied previously in [73]. In [73], the author used a log-barrier method to improve the performance of the maximum likelihood (ML) objective function when only four measurements are available. Here in this chapter, the performance of PSO is tested without the log-barrier.

Except for the case that the channel parameters may be unavailable, another problem for RSS based target localization is its application in distributed wireless sensor networks (WSNs). The positioning algorithms need to be extended for distributed processing. Distributed processing can help the network preserve energy by avoiding long distance wireless transmission. And it is more robust against communication time delay and changes in networks' topology due to mobility or node failure. The problem of distributed localization using acoustic energy measurements has been treated in [74], where a distributed algorithm based on projections onto convex sets is presented. The algorithm is shown to asymptotically approach the maximum likelihood estimate as the number of nodes increases when the target lies in the convex hull defined by the node's

coordinates. An incremental optimization algorithm is proposed in [75] for distributed maximum likelihood estimation. This method is applied to energy based localization. Simulation shows that its accuracy is only somewhat lower than that of the centralized method. In [76], the author also introduced an incremental gradient optimization method for energy based localization in WSN. This incremental gradient optimization method has low communication cost with an acceptable accuracy. In [77], a consensus based distributed algorithm has been used to localize a target while the energy measurements follow a contaminated Gaussian distribution. Each nodes collects measurements from adjacent nodes and compute an initial, then the initial is passed to adjacent nodes for consensus. Finally, based on the consensus in [31], the local estimates somewhat converge to the global estimate. In [30], the RSS based target localization problem is relaxed into a semidefinite programming (SDP) problem, and further solved by a consensus based distributed SDP [31]. It is shown that the distributed algorithm has the ability to converge onto the centralized solution, but the gap between Cramer Rao Lower Bound (CRLB) and the distributed algorithm is still big.

In this chapter, the distributed PSO introduced in [78] to deal with bearing estimation problem is adapted to solve the distributed RSS based localization problem. In this method, each sensor node calculates the target position using its own local ML objective function, and passes its estimate to its adjacent nodes. Through consensus, nodes can agree to compute some desired quantity.

The remainder of this chapter is organised as follows. In Section 4.2, the problem formulation is presented and the ability for PSO to solve the problem is examined. In Section 4.3, the problem of distributed location estimation is discussed. The implementation of distributed PSO for location estimation in a distributed WSN is presented in Section 4.4. Section 4.5 evaluates the performance of the proposed algorithm.

## 4.2 Problem Formulation

Assuming the PLE $\alpha$ is an unknown parameter, the ML estimator to estimate the location of the target based on the path loss model is

$$(\hat{\mathbf{u}}, \hat{\alpha}) = \arg\min_{(\mathbf{u},\alpha)} \sum_{i=1}^{N} (P_i - P_0 + 10\alpha \log_{10} d_i)^2. \tag{4.1}$$

As can be seen in (4.1), the ML estimator is non-linear and non-convex, and thus hard to

solve, especially when $\alpha$ is unknown. The reason that (4.1) is non-convex is explained as follows. The second derivative for an arbitrary function $f^2(x)$ is $2f'^2(x) + 2f(x)f''(x)$. Assume $f(x) = \log(x)$, at $x = 10$, we have $2f'^2(x) + 2f(x)f''(x) < 0$, thus (4.1) is not convex. Iterative optimisation employing LS [25], SDP [23], and maximum a posteriori MAP [26] can be applied to optimise the cost function; however, initialisation problems remain. An alternative is to apply global optimization techniques, such as the numerous stochastic optimization algorithms including genetic algorithm (GA), PSO, simulated annealing (SA) etc., to conduct the optimization. The PSO is a simple optimization technique introduced by Eberhart and Kennedy [79], and has been widely used in optimization problem. It is shown in [80], PSO is more computational efficient than GA in most tests. And it appears that PSO outperforms the GA when used to solve unconstrained nonlinear problems. Thus, in this work, PSO is applied to optimize (5.18) to determine the solution. The PSO algorithm is briefly introduced in Appendix E.

Now, we examine the ability of PSO to optimise the cost function (4.1). A 20 m by 20 m area was considered for the simulation. Eight nodes were placed at fixed positions as the same as in Fig.3.1. A target node was placed at (6,14) m. The transmit power, $P_0$, the PLE, $\alpha$, and noise standard deviation values were 0 dBm and 4 and 1.5 dB, respectively. In the simulations, two scenarios are examined: both $P_0$ and $\alpha$ are known; both $P_0$ and $\alpha$ are unknown. The number of particles used in each scenario is 20 and 50 respectively. As there are four unknown variables in the cost function when $P_0$ and $\alpha$ are unknown, more particles are used to increase the search ability of PSO. The configuration of the parameters of PSO is shown in Table 4.1 on page 66.

The results of the simulations are shown in Fig.4.1. The left-hand column shows the cost function value convergence over 100 Monte Carlo runs in two scenarios. The other two columns show the histogram of the final converged value of the location estimates. It can be observed that the PSO algorithm has the ability to find an acceptable solution close to the target position in all scenarios. In Section 4.5, the PSO will be compared with CRLB and other existing algorithms.

## 4.3 Distributed Location Estimation

In this section, the target location estimation problem in a distributed WSN is formulated as distributed optimisation problems. After collecting the RSS measurements from adjacent nodes, each sensor node formulates its local ML cost function and then attempts to estimate target location individually by solving its own cost function. In order to derive a distributed solution, firstly, a

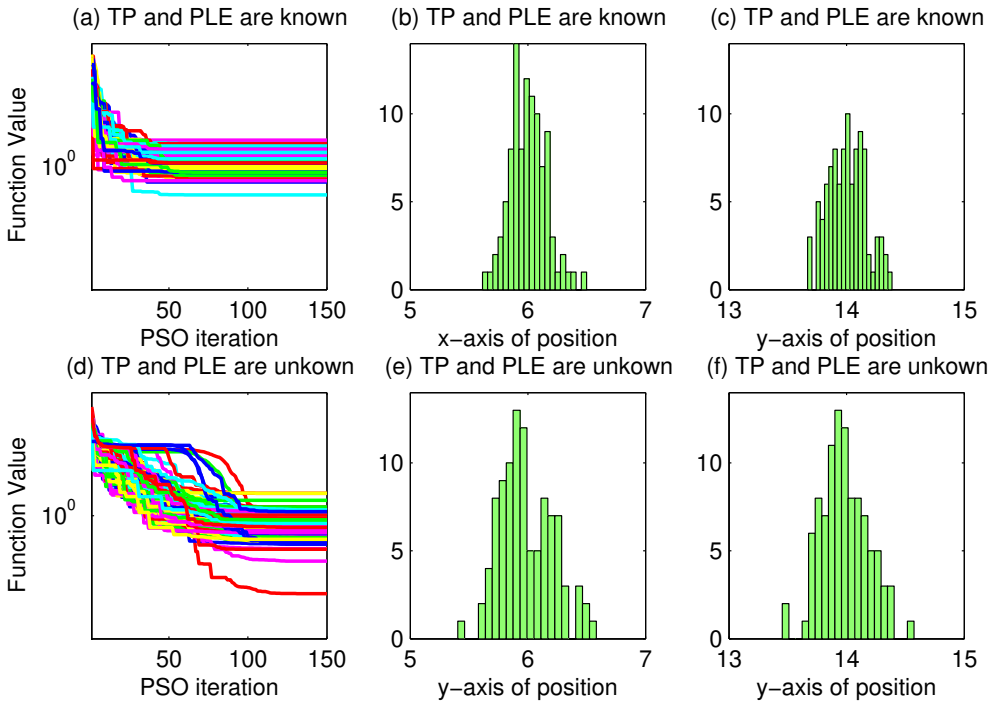



Figure 4.1: PSO performance in two scenarios. The left-hand column shows the cost function value convergence over 100 Monte Carlo runs in two scenarios. The other two columns show the histogram of the final converged value of the location estimates.

global consensus constraint is introduced into the cost function:

$$\begin{aligned} \underset{\mathbf{u},\{\mathbf{u}^{[i]}\}}{\text{minimise}} \quad & \sum_{i=1}^{N}(P_i - P_0 + 10\alpha \log_{10} d_i)^2 \quad i = 1, \ldots, N \\ \text{subject to} \quad & \mathbf{u}^{[i]} = \mathbf{u} \end{aligned} \tag{4.2}$$

where $\mathbf{u}^{[i]}$ is the local estimate of the target location at node $i$.

The problem is separable in the cost function, as it is the sum of $N$ squares. However, the global constraints are still not separable. In fact, in a distributed WSN, sensor nodes are more likely to only communicate with a node that is in their communication range. Thus, nodes can only agree with their adjacent nodes.

Let $\mathcal{N}_i$ be the set of adjacent nodes of node $i$. We can then reformulate the problem as follows:

$$\begin{aligned} \underset{\{\mathbf{u}^{[i]}\}}{\text{minimise}} \quad & \sum_{i=1}^{N}(P_i - P_0 + 10\alpha \log_{10} d_i)^2. \\ \text{subject to} \quad & \mathbf{u}^{[i]} = \mathbf{u}^{[j]}, j \in \mathcal{N}_i \end{aligned} \tag{4.3}$$

The global constraint that all local estimate have the same value in (4.2) is relaxed to a local constraint that the local estimate only need to agree with its adjacent nodes. Now, the constraint is also separable. If the WSN is connected, (4.3) and (4.2) are equivalent [81].

Then, the penalised problem [82] for (4.3) is given as follows:

$$\underset{\mathbf{u}^{[i]}}{\text{minimise}} \quad \sum_{i=1}^{N}(P_i - P_0 + 10\alpha \log_{10} d_i)^2 + c\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} \parallel \mathbf{u}^{[i]} - \mathbf{u}^{[j]} \parallel^2 . \tag{4.4}$$

where $c > 0$ is the penalty coefficient that controls the penalisation of the disagreement among adjacent nodes. In general, the choice of $c$ will influence the convergence rate.

As can be observed, (4.4) describes the centralised cost function as a sum of the local cost functions, which can be implemented in a distributed manner [83]. Based on the local and adjacent nodes' measurements, the local cost function is given as follows:

$$\underset{\mathbf{u}^{[i]}}{\text{minimise}} \quad \sum_{j\in\mathcal{N}_i}((P_i - P_0 + 10\alpha \log_{10} d_i)^2 + c \parallel \mathbf{u}^{[i]} - \mathbf{u}^{[j]} \parallel^2). \tag{4.5}$$

The local cost function (4.5) is non-convex as proved in Section 4.2 that the sum of squares of logarithm is nonconvex. However, all the local cost functions share common global minima if the noise is small enough that it can be ignored. If the local estimates get close enough to their global minima, the cost function is convex in that neighbourhood. Therefore, by using an iterative search algorithm initialised near the global minima, the distributed algorithm may converge to the global minima through a consensus method. PSO has the ability to reach a value close to the global maximum. Then a consensus algorithm can serve as a basic mechanism for distributing the optimization among the sensor nodes. By combining the PSO with a consensus algorithm, it enables the individual sensor node interact with other adjacent nodes to make the search process faster and accurate [78]. A distributed particle swarm optimisation (DPSO) method is proposed in [84], combining PSO with the consensus method. In DPSO, the node $i$ runs the PSO algorithm to optimise its own cost function and simultaneously shares its global best solution $\mathbf{g}^{[i]}$ with its adjacent nodes. Each sensor node starts to optimise its own local cost function using PSO and updates its estimate at iteration step $t$, $t = 1, 2, \ldots, T_{\text{max}}$, where $T_{\text{max}}$ is the maximum iteration number for PSO. Let $\mathbf{u}^{[i]}(t)$ be the location estimate at node $i$ at time $t$. During the update process, each sensor node combines its own current estimate $\mathbf{u}^{[i]}(t)$ with the estimates $\mathbf{u}^{[j]}(t)$ received from

its adjacent node $j$. The update equation is as follows [84]:

$$\mathbf{u}^{[i]}(t+1) = \sum_{j=1}^{N} \mathbf{W}_{ij}\mathbf{u}^{[j]}(t), \quad i = 1, \cdots, N, \tag{4.6}$$

where the $\mathbf{W}_{ij}$ is the impact factor of node $j$ on node $i$. In this work, the metropolis weight is employed. A brief description of the metropolis weight is provided in Appendix D.

## 4.4 Distributed PSO

In distributed location estimation, the question is how different nodes mutually share global best information with other nodes. To answer this, the DPSO algorithm [78] is used. In the DPSO, each node has its own particle swarm. At each iteration, each node updates its particles' positions, velocities, and best $\mathbf{p}_b^k$ using its local objective function, and shares its global best with adjacent nodes.

In sharing the global best, the consensus mechanism given in (4.7) is used to fuse the estimates received from adjacent nodes:

$$\mathbf{g}^{[i]}(t+1) = \sum_{j=1}^{N} \mathbf{W}_{ij}\mathbf{g}^{[j]}(t). \tag{4.7}$$

where the $\mathbf{g}^{[i]}$ is the global best of node $i$ at the $t$th iteration.

The fused $\mathbf{g}^{[i]}$ is used in the local optimisation process to estimate the global location, so that it can rapidly respond to changes in adjacent nodes. Further, the fused $\mathbf{g}^{[i]}$ becomes the current global best at each node. The computed global best information is exchanged between all participating nodes for further calculation.

The formulation of the DPSO algorithm for the ML estimator optimisation to estimate the location in the WSN is described as follows. In the beginning, every node in the network initialises a population of particles in the search space with random positions and random velocities, constrained by the length of the surveillance area. Every particle at each node is a candidate solution in the search space. Then, every particle is evaluated using the local ML cost function. The velocity of each node is updated using the inertia weight strategy (E.4) on page 111. At the

end of one iteration, each node shares its gbest with others. Then, each node updates its gbest and local ML cost function. Note that in the first iteration of DPSO, the gbest from adjacent nodes is unavailable; the constraint part in the local ML cost function is therefore set to zero.

The main steps in DPSO are summarised in Algorithm 4.4.1.

**Algorithm 4.4.1:** Distributed PSO for location estimation

Setup Problem:
- Initialise location of each node and set connection with adjacent nodes.
- Exchange RSS measurement, location, degree information between adjacent nodes.
- Define local ML cost function at each node and set the penalty coefficients.
- Define PSO parameters and problem search space.

Swarm initialisation at each node:
- Randomly initialise positions and velocities.

**for** *each iteration* **do**
  **for** *each node* **do**
    **for** *each particle* **do**
      Map particle position to solution vector in solution space;
      Evaluate the local cost function of current iteration of node according to (4.5);
      Update particle best location $\bar{\mathbf{p}}$ and group best position $\mathbf{g}^{[i]}$ according to fitness value;
      Update particle velocity according to (E.4) on page 111;
      **if** *velocity exceeds maximum* **then**
        Clamp particle velocity according to (E.3) on page 111;
      **end**
      Update particle position using (E.2) on page 110;
      **if** *particle position out of boundary* **then**
        set random location;
      **end**
    **end**
    Share global best position $\mathbf{g}^{[i]}$ with adjacent nodes;
  **end**
  Perform local diffusion, update local cost function (4.5);
  Check if maximum iteration number $T_{\max}$ is reached;
**end**

## 4.5 Numerical results

This section provides several Monte Carlo simulations to evaluate the performance of PSO in centralized networks and DPSO in distributed networks. The use of PSO has been studied previously in [73]. In their work, a log barrier is added in the cost function (4.1). However, the performance of PSO used on the original cost function (4.1) is not studied. In Section 4.5.1, the PSO algorithm is used to optimize (4.1) directly, and its localization accuracy is compared with

CRLB and other existing algorithms. In Section 4.5.2, the performance of DPSO is examined. In the following simulation scenarios, the unit for locations is metres. The RSS measurements are generated by adding zero-mean Gaussian noise to the true values.

### 4.5.1 Simulations of the centralized algorithm

The topology in Fig.3.1 is used for the following simulations. The simulated standard deviation of noise $\sigma$ ranges from 1 dB to 7 dB with 1 dB step size. One thousand Monte Carlo trials are performed for each value. The PSO control parameters are specified in Table 4.1. PSO is terminated after the maximum number of iterations is reached. The transmit power and the PLE are $-15$ dBm and 3, respectively, unless otherwise stated. The solution search spaces for the position, transmit power, and PLE are 0 m to 20 m, -30 dBm to 0 dBm, and 1.5 to 6.5, respectively.

| PSO parameters | | | | | |
|---|---|---|---|---|---|
| $\psi_1$ | $\psi_2$ | $t_{\max}$ | $P_s$ | $w_{max}$ | $w_{min}$ |
| 2 | 2 | 200 | 30 | 0.9 | 0.4 |

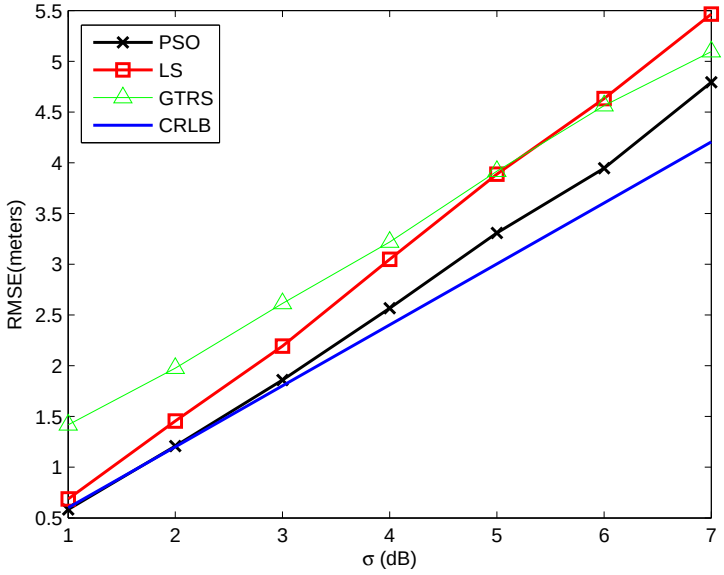
Table 4.1: PSO control parameters.

In the first simulation, the PLE is assumed to be unknown to the system, and the locations of the target and the PLE are jointly estimated by PSO. The PSO algorithm is compared with the CRLB, Generalized Trust Region Subproblem (GTRS) [51] and iterative LS [25]. The initial PLE value for both the GTRS and LS algorithms is 4.

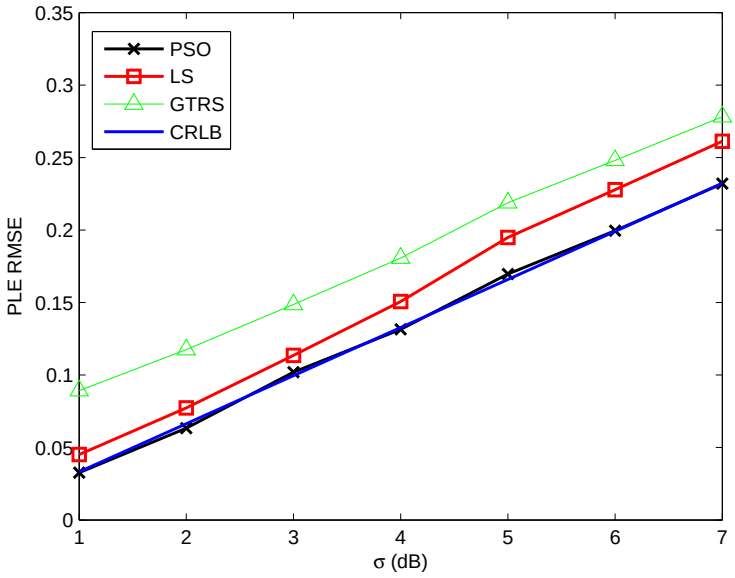
Fig.4.2a shows the RMSE of the location estimation. As the figure shows, the PSO algorithm outperforms the other algorithms, and can reach the CRLB in the whole range of $\sigma$. Fig.4.2b illustrates the RMSE of the PLE estimation for different approaches. As can be observed, the PSO method can still reach the CRLB and outperform other algorithms. One more advantage is that PSO does not need initialisation like other algorithms.

In the next simulation, the same experiment is repeated with transmit power also unknown. The PSO algorithm is compared with the GTRS algorithm [51] and an iterative LS algorithm [60].

30 particles works well when $P_0$ or $\alpha$ is unknown. However, when both of them are unknown, in the simulation, one can observe a gap between the CRLB and PSO. The parameters of PSO need to be adjusted for further tests. Empirical results have shown that the number of particles composing the swarm can influence the resulting performance depending on the problem [85].




(a) Location




(b) PLE

Figure 4.2: The RMSE of the PSO, LS, and GTRS algorithms, and the CRLB for (a) location estimation, and (b) PLE estimation.

Thus, the swarm size is adjusted to 50 which is recommended as the standards for PSO [85]. Simulation results are shown in Fig.4.3. The findings are similar to Fig.4.2a. PSO still has the best performance among all the methods.

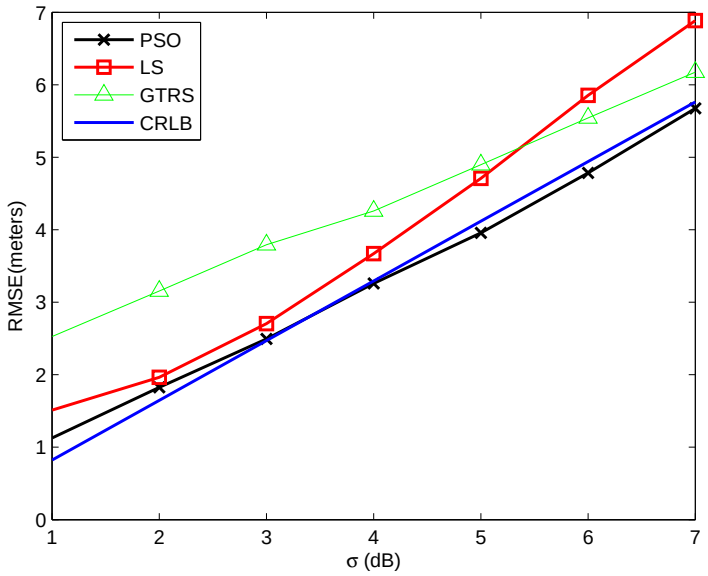



Figure 4.3: The RMSE of the PSO, LS, and GTRS algorithms and the CRLB for location estimation against the variance of shadowing, where transmit power and PLE are unknown.

### 4.5.2 Simulations of the distributed algorithm

In this section, a numerical example is presented to demonstrate the performance and iteration behaviour of the DPSO-based location estimation method.

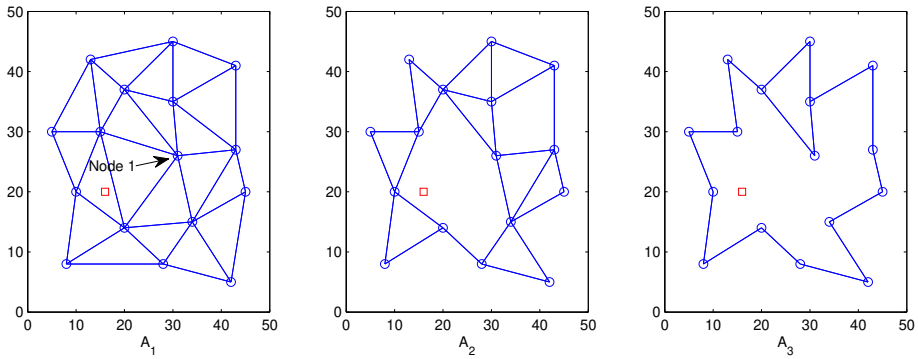



Figure 4.4: Network topology used in simulation.

First, the network is initialised such that each node is aware of its adjacent nodes' RSS measurements and positions, thus, each node can build its own local ML objective function. In this example, we consider a network containing 16 nodes deployed over an area of 50 m by 50 m.

The topology of the network is as shown in the left graph in Fig.4.4. The target is located at (16, 20)m. The same system parameters are used for all nodes with transmit power $P_0 = -15$ dB and PLE $\alpha = 4$. The DPSO parameters are shown in Table 4.1. The penalty coefficient is $c = 0.05$. The average RMSE (ARMSE) in the following simulations is calculated as follows:

$$\text{ARMSE} = \sqrt{\frac{1}{NM_{run}} \sum_{l=1}^{M_{run}} \sum_{i=1}^{N} ||\hat{\mathbf{u}}^{[i]}(l) - \mathbf{u}||}. \tag{4.8}$$

In the first simulation, it is assumed that the PLE and transmit power are known *a priori*.

Fig.4.5 shows the mean RMSE of the location estimate for DPSO and distributed weighted SDP (DwSDP) [30] against the variance of shadowing. For comparison, the RMSE of the Global PSO, Global wSDP, and DPSO algorithms at node 1, and the corresponding CRLB are also included. All simulations in Fig.4.5 include 1000 Monte Carlo simulations.

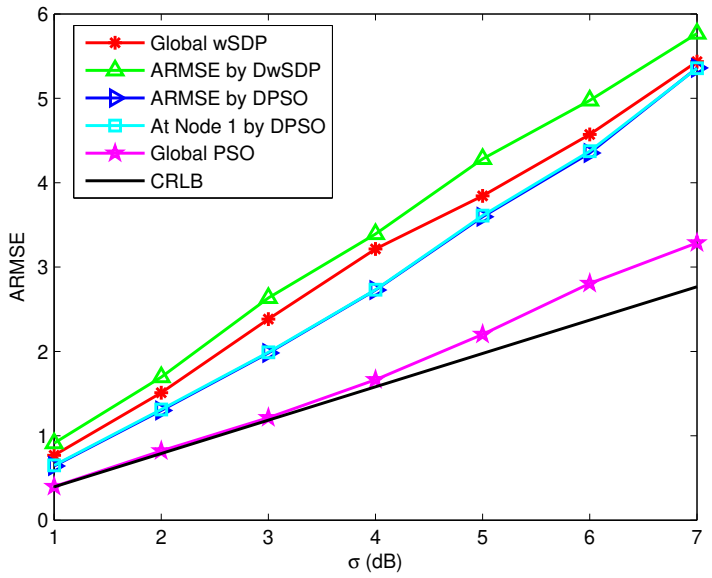



Figure 4.5: The RMSE of the Global PSO, DPSO, and DwSDP algorithms and the CRLB for location estimation against the variance of shadowing.

As expected, Global PSO has the best performance. wSDP failed to reach the CRLB because a biased noise was introduced during relaxation. It can also be observed that DPSO outperforms DwSDP over the whole range of shadow noise. It is also noticeable that the mean RMSE of DPSO has the same performance as DPSO at node 1, which implies that the estimates at each node converge. Although there is a gap between the DPSO and Global PSO values, the performance of the proposed method seems to be acceptable.

The convergence behaviour of the distributed algorithm with shadowing noise $\sigma = 3$ is depicted in Fig.4.6. Net 1 in the figure refers to a small network combined with node 1 and its adjacent nodes. Each line represents the average RMSE over 300 independent iterations. As can be observed, all the algorithms have similar convergence speed and converge within 20 iterations.

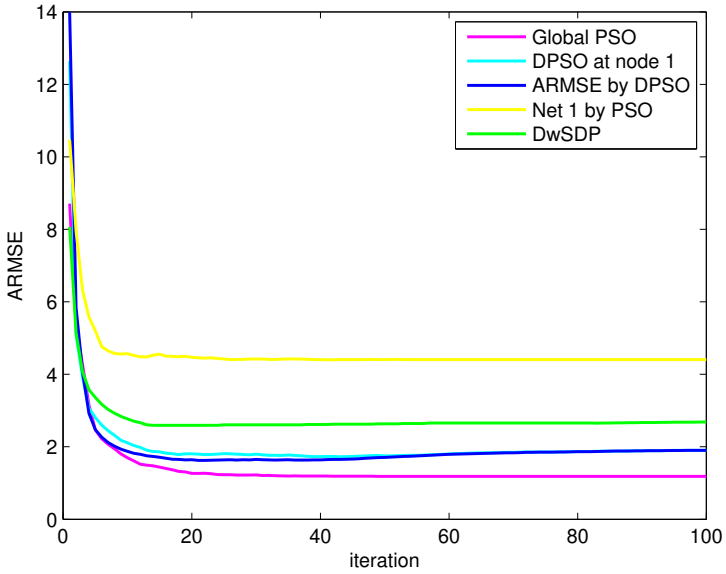



Figure 4.6: ARMSE variation for Global PSO, DPSO, and DwSDP methods.

The topology used above is highly connected with algebraic connectivity $\lambda_2(A_1) = 0.932$. In the next simulation, we test the DPSO algorithm in other two topology with smaller algebraic connectivity. The nodes are placed on the same position, but the topology is different. Their topologies are shown as $A_2$ and $A_3$ in Fig.4.4 with algebraic connectivity $\lambda_2(A_2) = 0.419$ and $\lambda_2(A_3) = 0.198$. Their localization accuracy in the sense of ARMSE is plotted in Fig.4.7. As expected, as the algebraic connectivity decreases, the localization accuracy also declines. Their convergence behaviour is also depicted in Fig.4.8. The DPSO in topology $A_3$ has the lowest convergence speed, because it is less connected than the other two topologies, thus the information exchange in $A_3$ is slow.

The same experiment shown in Fig.4.5 is illustrated in Fig.4.9 when both transmit power and PLE are unknown. The PSO control parameters used in this simulation are the same as the previous, except that the iteration number is increased to 200. Also, the particle number for Global PSO is increased to 50 to achieve better performance. Since DwSDP cannot solve the problem when transmit power and PLE are unknown, it is not included in the following figures.

It can be observed in Fig.4.9 that as shadow noise variance increases, the gap between Global PSO

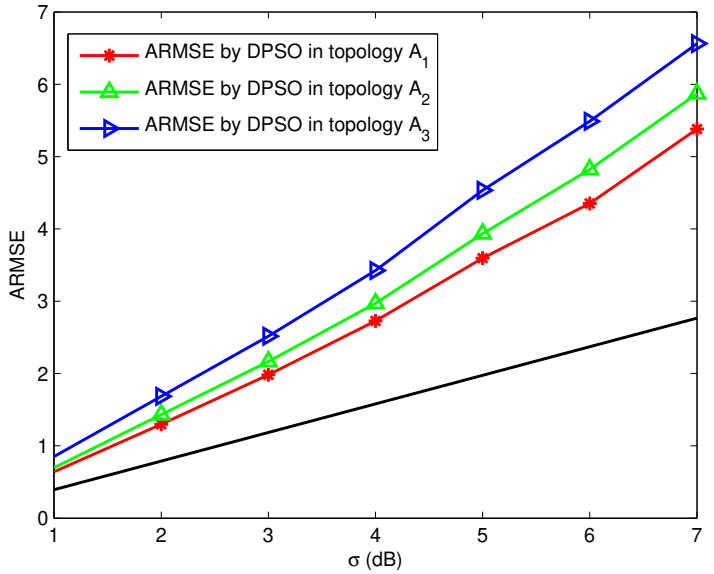



Figure 4.7: The ARMSE of the DPSO in different topologies.

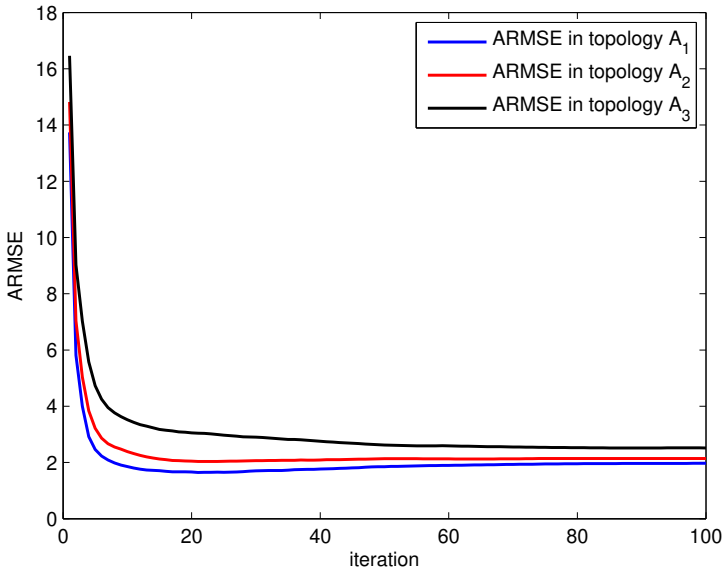



Figure 4.8: ARMSE variation for DPSO in different topologies.

and the CRLB tends to become larger though it remains acceptable. As expected, there is also some degradation in the performance of DPSO in comparison to Global PSO.

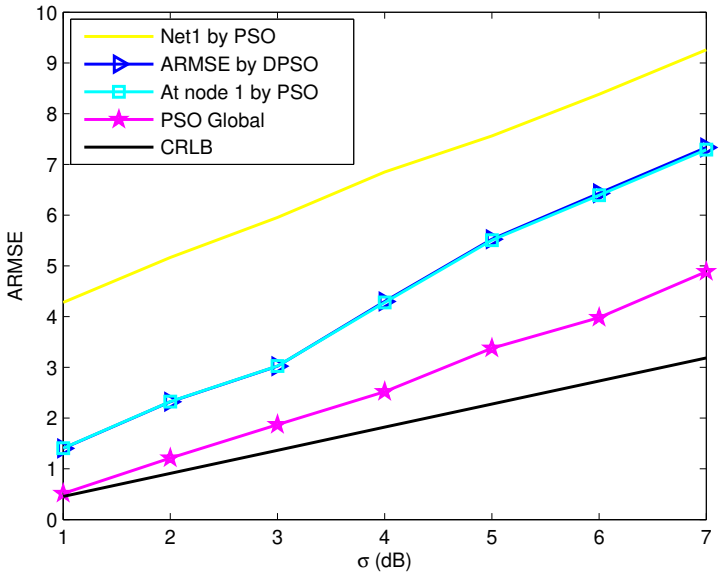



Figure 4.9: The RMSE of the Global PSO and DPSO algorithms and the CRLB for location estimation against the variance of shadowing when transmit power and PLE are unknown.

Finally, we compare the convergence behaviour of the distributed algorithm against the global one in relation to shadowing noise $\sigma = 3$ in Fig.4.10. Interestingly, DPSO converges faster than Global PSO. A reasonable explanation is that DPSO has only 30 particles at each node, but can communicate with its adjacent nodes. Through consensus, the particles at each node have a greater chance of finding a better solution than with Global PSO, where there are 50 particles at the start of the iterations. This is similar to multi swarm optimisation [86] where multiple sub-swarms are used instead of a bigger swarm. This gives PSO a better chance of finding an optimal solution.

## 4.6 Summary

PSO is used to optimise the ML objective function of the RSS-based location estimation. The use of PSO has been studied previously in [73]. In their work, a log barrier is added in the cost function (4.1). However, the performance of PSO used on the original cost function (4.1) is not studied. The simulation results show that PSO has the ability to reach the CRLB when the PLE is unknown. Unlike other existing methods, PSO does not require an initial PLE. By combining PSO with a consensus algorithm, the estimation problem can be solved in a distributed manner in distributed WSNs. Then, the DPSO proposed in [78] for bearing estimation is used to localize the

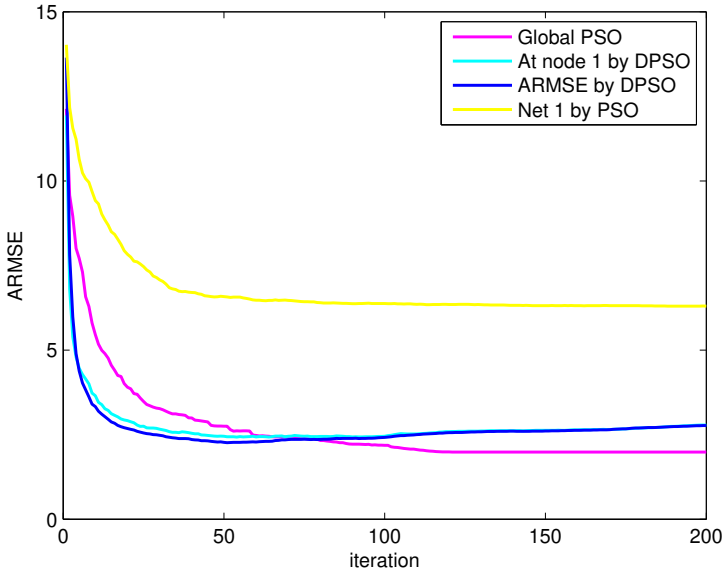



Figure 4.10: RMSE variation for Global PSO and DPSO methods when transmit power and PLE are unknown.

target location distributedly. Simulation results have demonstrated that the performance of DPSO in RSS based target localization is acceptable in terms of RMSE. More importantly, it has the ability to locate the target when the PLE and transmit power are unknown while the distributed SDP cannot.

# Chapter 5

# Distributed Target Localization Using QRSS Measurements

## 5.1 Introduction

As mentioned in previous chapters, sensor nodes in wireless sensor networks (WSNs) are constrained by energy and communication bandwidth. To save energy, the Quantized RSS (QRSS) model is proposed in [32]. There are many quantization scheme, such as the uniform quantization method and the vector quantization method. In any quantization scheme, thresholds are the most important parameters, especially when a small number of bits are used. In [87], a heuristic method to determine the optimum quantization thresholds for target localization using quantized acoustic energy measurement is presented. In [88], target localization using quantized measurements considering the wireless channel statistics is presented. Target localization using quantized data combined with coding is also presented in [89]. The well known proximity measurement [61], [64] localization can be considered as a special case of QRSS model where only one bit data is used.

Another way to save energy is to use distributed algorithms to avoid long distance data transmission as mentioned in Section 4.1. Distributed iterative search techniques such as distributed gradient method, distributed Gauss-Newton method and distributed expectation maximization (EM) can also be applied to optimize the maximum likelihood (ML) function in a distributed manner. However, these techniques have limitations because of the multimodal nature of the ML function. In this work, a distributed localization method consisting of two steps is proposed. In the first step, each sensor node exchanges its QRSS measurement with adjacent nodes and then estimate the target position by optimizing a local ML function. If the initialization of the local ML function is in the near vicinity of the global minimum, where the local ML is approximately convex, the distributed algorithm such as distributed gradient and distributed Gauss-Newton method can use the estimate as an initialization and take over from there. Here

in this chapter, the PSO is used for initialization, as it can handle multimodal cost functions and has been employed to optimize the ML function for RSS-based localization problem [73]. To save communication, the data exchanged in the distributed process also needs to be quantized. The quantized distributed gradient (QDG) [90] is applied to conduct the process as only the target position needs to be exchanged between adjacent nodes in distributed gradient algorithm. Moreover, the target position is easy to quantize, as the target is located in an area of interest and the range of the area can be known to the system. Unlike the distributed gradient algorithm, the dynamic range of the intermediate parameters exchanged in algorithms such as distributed Gauss-Newton [91] is difficult to acquire and is thus challenging to quantize. An improved QDG algorithm with less quantization bits is also proposed in the paper. The performance of the distributed algorithms is compared with the centralized PSO-ML algorithm via simulation. Note that the RSS measurements and the local estimate at sensor node are quantized in different ways.

The rest of this chapter is organized as follows. In Section 5.2, the problem formulation with the RSS and QRSS signal model is first introduced. In Section 5.3, the quantization thresholds are designed, and the ML problem using quantized data are solved using PSO. In Section 5.4, we propose the distributed algorithm using a gradient method with a quantization error compensation term. Simulation results are provided in Section 5.5. Concluding remarks are given in Section 5.6.

## 5.2 Signal Model and Problem Formulation

When quantized measurements are used in the network, especially when quantization level is low, more nodes are required to achieve the same localization accuracy when unquantized measurements are used. Therefore, the network size (number of nodes) used in this chapter is larger than the one used in previous chapters. Consider a wireless sensor network, as illustrated in Fig. 5.1, consisting of $N$ sensor nodes deployed on a certain area. Nodes are static and able to communicate with adjacent nodes that lie within a given range. Assume that a target emits signals which can be heard by all nodes in the network. The goal is to determine the location of the target.

For estimating the location of the target, we employ the QRSS measurements. Denote the quantized version of $P_i$ as $K_i, (i = 1, \ldots, N)$, where $K_i$ can take any discrete value from 0 to $L = 2^M - 1$. For simplicity, we assume all the sensor nodes employ the same quantization threshold. In the quantization process, with the quantization thresholds $\mathbf{s} = [s_0, s_1, \ldots, s_L]$, the

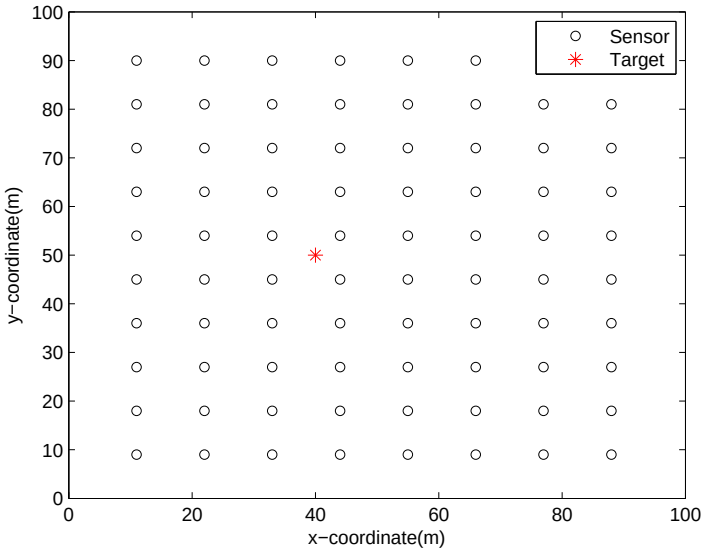



Figure 5.1: A target in sensor network with 80 sensor nodes.

raw RSS measurements are quantized into discrete data $K_i$

$$K_i = \begin{cases} 0 & \text{if } s_0 < P_i < s_1 \\ 1 & \text{if } s_1 < P_i < s_2 \\ \vdots & \vdots \\ L-1 & \text{if } s_{L-1} < P_i < s_L \end{cases}, \tag{5.1}$$

where $s_0 = -\infty$ and $s_L = \infty$. The QRSS measurement $K_i$ will be used to estimate the target position.

## 5.3 Localization Strategies

In this section, we present localization strategies using the QRSS measures at the nodes. We first consider the centralized ML estimate and then the ML estimate will be used to calculate an initial value for the distributed method described in the next section.

### 5.3.1 Maximum Likelihood Estimation

Using the RSS measurement model (2.4) and the quantization method (5.1), under the Gaussian shadowing noise assumption, the probability that node $i$ takes a decision $K_i$ is

$$p_i(K_i|\mathbf{u}) = \begin{cases} Q(\frac{s_0-z_i}{\sigma}) - Q(\frac{s_1-z_i}{\sigma}) & K_i = 0 \\ Q(\frac{s_1-z_i}{\sigma}) - Q(\frac{s_2-z_i}{\sigma}) & K_i = 1 \\ \vdots & \vdots \\ Q(\frac{s_{L-1}-z_i}{\sigma}) - Q(\frac{s_L-z_i}{\sigma}) & K_i = L-1 \end{cases}, \tag{5.2}$$

where $z_i$ is defined in (2.9), $Q(\cdot)$ is defined in (2.75) and $\mathbf{u}$ is the target location.

Consider the presence of a certain unit that gathers all measurements coming from the sensor nodes. After collecting data $\mathbf{K}$, where $\mathbf{K} = [K_1, \ldots, K_N]^T$, the centre unite estimates the parameter vector $\mathbf{u} = [x, y]^T$. Based on the notations and assumptions, it is easy to derive the likelihood function at the center unit

$$p(\mathbf{K}|\mathbf{u}) = \prod_{i=1}^{N} p_i(K_i|\mathbf{u}), \tag{5.3}$$

where the product of probabilities is due to the assumption of independent QRSS measurements.

Since the logarithm is a monotonically increasing function, and by taking logarithm of (5.3), multiplication becomes addition, so that (5.3) is separable. The logarithm of (5.3) is

$$f(\mathbf{K}|\mathbf{u}) = \sum_{i=1}^{N} \ln[p_i(K_i|\mathbf{u})]. \tag{5.4}$$

Therefore, the ML estimator, $\mathbf{u}$, is now the solution of the following maximization problem

$$\hat{\mathbf{u}} = \arg\max_{\mathbf{u}} f(\mathbf{K}|\mathbf{u}). \tag{5.5}$$

Fig.5.2 is an illustration of how the cost function looks like for a network of 40 nodes deployed in the surveillance area. It is clear that the function has several minima and saddle point may exist. As can be seen in the figure, the cost function (5.5) is not linear nor convex. These problems are

mostly overcome in PSO-based solutions.

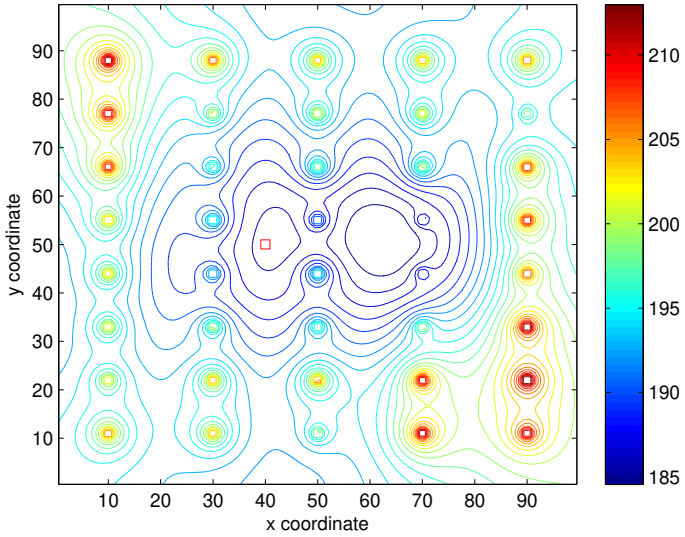



Figure 5.2: Contour plot of the ML objective function. The red square represents the target location.

Notice that the only parameters the center unit needs to know to locate the target are sensor nodes locations, quantization threshold and the RSS model, i.e., $P_0$, $\alpha$, $\sigma^2$. The method to calculate $P_0$, $\alpha$, $\sigma^2$ is described in Section 2.1.

### 5.3.2 Threshold Design Method

Before solving the optimization problem in (5.5), we must determine a set of quantization thresholds $\mathbf{s}$ for RSS measurements. As reviewed in Section 2.3, a natural way to choose the optimal threshold is to minimize the location estimation errors in $\hat{\mathbf{u}}$ with respect to the thresholds $\mathbf{s}$. In other words, minimizing the right hand side of (2.17) gives the optimal threshold. However, as shown in (2.78), $\mathbf{J}_{11}$, $\mathbf{J}_{12}$, $\mathbf{J}_{22}$ are all functions of the target location which begs the question because it is the target location that must be estimated. Furthermore, the Fisher information matrix (FIM) $\mathbf{J}$ also contains parameters like the locations of the sensor nodes. For many WSNs, the sensors will be deployed randomly in the surveillance area. Due to the uncertainty of nodes positions, it is difficult to set the thresholds before deployment. A possible solution is to assume that the target position $\mathbf{u}$ and sensor position $\mathbf{c}_i$ follow a uniform distribution in an interval in the surveillance area [87]. Then we can calculate the probability density function (PDF) of the received signal strength $z_i$ at a random location in the surveillance area. Using this pdf, the thresholds can be determined [87].

As no prior information on target location is provided, it is reasonable to assume that $x_i, y_i, x, y$ are i.i.d and follow a uniform distribution in the interval $[-b/2, b/2]$. Denote $v_i$ as the square distance between node $i$ to the target. As shown in [87], the pdf of $v_i$

$$f(v_i) = \begin{cases} \frac{\pi}{b^2} + \frac{v_i}{b^4} - \frac{4\sqrt{v_i}}{b^3} & 0 < v_i \leq b^2 \\ \frac{2}{b^2}\arcsin(\frac{2b^2 - v_i}{v_i}) - \frac{v_i}{b^4} + \frac{4\sqrt{v_i - b^2}}{b^3} - 2/b^2 & b^2 < v_i \leq 2b^2 \\ 0 & \text{otherwise} \end{cases}. \tag{5.6}$$

The probability that $v_i$ is greater than 1 meter is [87]:

$$\gamma = 1 - \int_0^1 f(v_i)dv_i \tag{5.7}$$

$$= 1 + \frac{8}{3b^3} - \frac{\pi}{b^2} - \frac{1}{2b^4}. \tag{5.8}$$

Hence if $v_i$ is greater than or equal to 1 meter, we have

$$f_v(v_i|v_i \geq 1) = \frac{1}{\gamma} f_v(v_i). \tag{5.9}$$

With transmit power $P_0$ and path loss exponent $\alpha$ known *a prior*, using the probability transformation rule, the pdf of the received signal strength $z_i$ at a random location in the surveillance area is

$$f_z(z_i) = \frac{\ln 10}{5\alpha} g_i \gamma \begin{cases} \frac{\pi}{b^2} + \frac{g_i}{b^4} - \frac{4\sqrt{g_i}}{b^3} & \phi_1 < z_i \leq P_0 \\ \frac{2}{b^2}\arcsin(\frac{2b^2 - g_i}{g_i}) - \frac{g_i}{b^4} + \frac{4\sqrt{g_i - b^2}}{b^3} - \frac{2}{b^2} & \phi_2 < z_i \leq \phi_1 \\ 0 & \text{otherwise} \end{cases}, \tag{5.10}$$

where

$$g_i = 10^{(P_0 - z_i)/(5\alpha)}, \tag{5.11}$$

$$\gamma = 1 + \frac{8}{3b^3} - \frac{\pi}{b^2} - \frac{1}{2b^4}, \tag{5.12}$$

$$\phi_1 = P_0 - 5\alpha \log_{10} b^2 \tag{5.13}$$

and

$$\phi_2 = P_0 - 5\alpha \log_{10} 2b^2. \tag{5.14}$$

Note that the RSS model is different from the one used in [87]. So the pdf (5.10) is different from the work in [87]. Simulation is carried out to support the results in (5.10). In Fig.5.3, the red solid line represents the PDF derived in (5.10). And the blue bar represents the normalized number of samples generated using the definition of $x$, $x_i$, $y$ and $y_i$. Obviously, the PDF of the received signal strength $z$ generated by simulation is identical to (5.10).

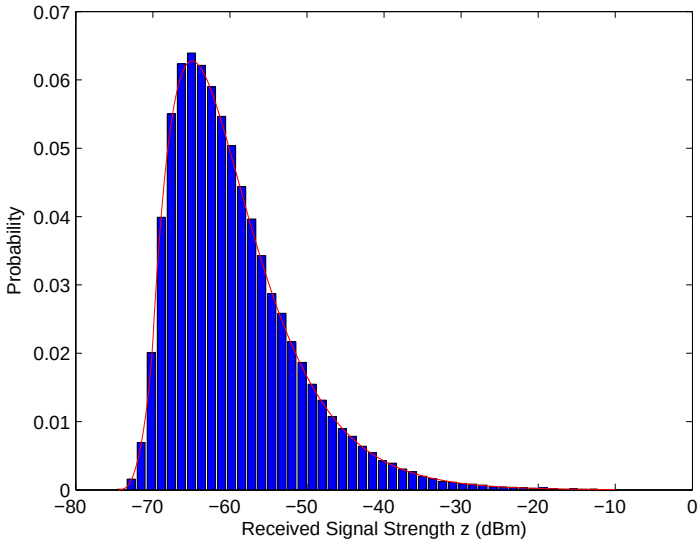



Figure 5.3: PDF of received signal strength $z$ in surveillance area ($\alpha = 3$, $P_0 = -10$dBm, $b = 100$m, 100000 samples).

All the information about the target position is contained in received signal strength $z_i$. If all the RSS measurements can be restored from the QRSS measurement precisely, the target position can be estimated accurately. Therefore, the optimum quantization threshold can be chosen so that in QRSS measurement $K$, there is maximum information about $z_i$. Similar to the derivation of (5.4), the log-likelihood function of QRSS data $K$ received at any sensor is $\ln p(K|z_i)$, where

$$p(K|z_i) = \begin{cases} Q(\frac{s_0 - z_i}{\sigma}) - Q(\frac{s_1 - z_i}{\sigma}) & K = 0 \\ Q(\frac{s_1 - z_i}{\sigma}) - Q(\frac{s_2 - z_i}{\sigma}) & K = 1 \\ \vdots & \vdots \\ Q(\frac{s_{L-1} - z_i}{\sigma}) - Q(\frac{s_L - z_i}{\sigma}) & K = L - 1 \end{cases}. \tag{5.15}$$

Then the Fisher information with respect to $z$ of the threshold estimation problem is

$$\begin{aligned} J_z &= -E[\frac{\partial^2 \ln p(K|z_i)}{\partial z_i^2}] \\ &= \sum_{K=0}^{L-1} \frac{[\exp(-\frac{(s_l - z_i)^2}{2\sigma^2}) - \exp(-\frac{(s_{l+1} - z_i)^2}{2\sigma^2})]^2}{2\pi\sigma^2 p(K|z_i)}, \end{aligned} \tag{5.16}$$

The derivation of (5.16) is shown in Appendix G. Then, the average Fisher information $z_i$ have in the surveillance area is

$$F(\mathbf{s}) = \int_{\phi_2}^{P_0} J_z f(z_i) dz_i. \tag{5.17}$$

Now, maximizing (5.17) with respect to $\mathbf{s}$ could solve the threshold design problem

$$\arg\max_{\mathbf{s}} F(\mathbf{s}). \tag{5.18}$$

When $M = 1, 2$, (5.18) can be solved using a method like grid search or genetic algorithm. When $M$ becomes larger, finding a maximum in $2^M - 1$ dimension is difficult. By using a uniform quantization threshold, the high dimensional optimization problem could be simplified to a two parameter searching problem: the second threshold $s_1$ and the uniform quantization step size $\Delta = s_{l+1} - s_l$, for $l = 1, \ldots, L - 2$.

Note that the only parameter the system needs to know is the size of the surveillance area. The thresholds for RSS measurement can be determined off-line.

## 5.4 Distributed Algorithm

In this section, the quantized distributed gradient (QDG) method is described. Further, an improved version of QDG is proposed.

As it can be seen, (5.4) describes the centralized cost function as a sum of local cost functions, which can be implemented in a distributed manner [92]. If the local ML function

$$f_i = \ln p_i(K_i|\mathbf{u}) \tag{5.19}$$

is convex, then distributed optimization methods using consensus algorithm will converge to global optimum [93]. In a multimodal functions, the PSO algorithm can search the region where the global solution lies. With local measurement and those from the adjacent nodes, the sensor node can compute an initialization near the minimum. When the initialization of the local ML function is in the near vicinity of the global minimum, where the local ML is approximately convex, the distributed gradient method can take over and solve the problem. Here, we describe a distributed implementation of the ML cost function (5.5). The proposed algorithm involves two phases: QRSS measurement sharing phase and the distributed estimation phase. In the first phase, each node exchanges its quantized measurement data with its adjacent nodes. Assume each node knows the positions of its adjacent nodes, with at least 3 measurements, an initialization can be found using PSO. In the second phase, at each iteration $t$, the location estimate $\hat{\mathbf{u}}_i(t)$ at node $i$ is quantized and then transmitted to its adjacent nodes for update. Eventually, target location estimate at every node will converge to some extent close to the global optimum. Note that DPSO introduced in chapter 4 can also solve the problem. However, distributed gradient is simpler and faster than the DPSO, we will compare their performance in next section.

Different quantization methods could be applied to locally estimate $\hat{\mathbf{u}}(t)$, i.e. uniform quantization, exponential quantization [94]. Without loss of generality, we use uniform quantization, since it is easy to implement in hardware. In the case of $n$-bit quantizer, the uniform quantized values can be expressed as

$$x^Q = \lfloor \frac{x - x^{\min}}{\Delta} \rfloor \Delta + \Delta/2 + x^{\min}, \tag{5.20}$$

where $x^{\min}$ and $x^{\max}$ represents the minimum and maximum values of the quantization threshold. The parameter $\Delta = (x^{\max} - x^{\min})/2^n$ is the uniform quantization step size, which drives the quantization error. Since we have no prior information of the target location, we set $x^{\min}$ and $x^{\max}$ as the minimum coordinate and maximum coordinate of the surveillance area, i.e. $(-b/2 \quad b/2)$.

Define the quantized estimate of the target position at node $i$ as $\hat{\mathbf{u}}^{[i],Q}$. At the $t$th iteration, node $i$ computes an update according to

$$\hat{\mathbf{u}}^{[i]}(t+1) = \sum_{j=1}^{N} \mathbf{W}_{ij} \hat{\mathbf{u}}^{[j],Q}(t) - \beta \boldsymbol{\zeta}_i(t), \tag{5.21}$$

where $\beta > 0$ is a diminishing step size and $\boldsymbol{\zeta}_i(t)$ denotes the gradient of local cost function $f_i(\mathbf{u})$ at $\mathbf{u} = \hat{\mathbf{u}}^{[i]}(t)$. $\mathbf{W}$ is the Metropolis Weights introduced in Appendix C. Equation (5.21) is referred as Quantized Distributed Gradient (QDG-I) in the simulations. The gradient of $f_i$ in equation (5.19) is shown in Appendix H. The QDG algorithm is built under the assumption that the network follows a synchronous communication protocol. The asynchronous protocol requires the gossip algorithm [95], which is out of the scope of this thesis.

The quantized communication effect in distributed gradient method has been studied in [90]. When the initialization is near the vicinity of the global minimum, the quantized distributed gradient algorithm could converge to the optimal objective value within some error which depends on the number of quantization level.

In order to reduce the quantization error introduced in (5.20), we utilize the local unquantized data [96]. The update in each iteration becomes

$$\hat{\mathbf{u}}^{[i]}(t+1) = \sum_{j=1}^{N} \mathbf{W}_{ij} \hat{\mathbf{u}}^{[j],Q}(t) + \hat{\mathbf{u}}^{[i]}(t) - \hat{\mathbf{u}}^{[i],Q}(t) - \beta \boldsymbol{\zeta}_i(t). \tag{5.22}$$

This algorithm is referred as QDG-II in the simulations. To better understand how (5.22) works, following the derivation in [96], compact (5.22) into matrix form

$$\hat{\mathbf{U}}(t+1) = \mathbf{W}\hat{\mathbf{U}}^{Q}(t) + \hat{\mathbf{U}}(t) - \hat{\mathbf{U}}^{Q}(t) - \beta \mathbf{D}(t). \tag{5.23}$$

$\hat{\mathbf{U}}^{Q} = [\hat{\mathbf{u}}^{[1],Q}, \ldots, \hat{\mathbf{u}}^{[N],Q}]$ and $\mathbf{D}(t) = [\boldsymbol{\zeta}_1(t), \ldots, \boldsymbol{\zeta}_N(t)]$. Define $\mathbf{e}(t) = \hat{\mathbf{U}}^{Q}(t) - \hat{\mathbf{U}}(t)$ as the quantization noise in $\mathbf{U}^{Q}(t)$. Further, (5.23) can be rewritten as

$$\hat{\mathbf{U}}(t+1) = \mathbf{W}\hat{\mathbf{U}}(t) - (\mathbf{I} - \mathbf{W})\mathbf{e}(t) - \beta\mathbf{D}(t), \tag{5.24}$$

Expanding (5.24), we have

$$\hat{\mathbf{U}}(t+1) = \mathbf{W}^t\hat{\mathbf{U}}(1) - \sum_{k=1}^{t-k+1} \mathbf{W}^{t-k}(\mathbf{I} - \mathbf{W})\mathbf{e}(k) - \beta \sum_{k=1}^{t-k+1} \mathbf{W}^{t-k}\mathbf{D}(k). \tag{5.25}$$

It has been proved in [97],

$$\lim_{t\to\infty} \mathbf{W}^t = \frac{1}{N}\mathbf{1}\mathbf{1}^T. \tag{5.26}$$

For certain $k$,

$$\lim_{t\to\infty} \mathbf{W}^{t-k+1}(\mathbf{I} - \mathbf{W})\mathbf{e}(k) = 0. \tag{5.27}$$

As shown in (5.27), when the iteration process goes on, the quantization error introduced in the $k$th iteration and before $k$th iteration gradually vanishes. However, recent quantization noise remains.

The distributed localization algorithm discussed above can be summarized in the following.

1. Initialize the WSN with system parameters, $b$, $P_0$, $\alpha$ and $\sigma$. Determine the thresholds for RSS measurements by optimizing (5.18) using PSO.
2. The $i$th node broadcasts its QRSS measurements to all neighboring nodes. It also receives the broadcasts from its neighbors.
3. The $i$th node calculates the initial value $\hat{\mathbf{u}}_i(1)$ using PSO.
4. The $i$th node quantizes its current estimate $\hat{\mathbf{u}}_i(t)$ using (5.20) and broadcasts it to its adjacent nodes and uses the quantized distributed gradient algorithm (5.21) or (5.22) to update the local estimates until the maximum number of iteration is reached.

## 5.5 Numerical results

In this section, we simulate several scenarios for the threshold optimization, the centralized formulation (5.5) and the distributed algorithms (5.21), (5.22). In the simulations for every realization, the transmit power $P_0$, and the path loss exponent $\alpha$ are $-10$dBm and 3, respectively. The above parameters are also used to design the quantization threshold using the method presented in section 5.3.

### 5.5.1 Threshold simulations

In this section, threshold value are calculated using PSO. Its control parameters are given in Tab.4.1. The results for 1-bit and 2-bit quantization are shown in Fig.5.4. The left column shows the cost function value convergence over 100 MC trials while 1-bit or 2-bit quantization is employed. And the right column shows the final converged value of the threshold estimates for each of the trials. It can be seen that in both cases, PSO can always converge to global maxima and find the thresholds. According to the simulations, the threshold for 1-bit quantization is $-60.32$dBm and the thresholds for 2-bit quantization is $[-65.78 - 58.33 - 49.85]$dBm. For quantization bits more than 2, samples are generated using (5.17), then (5.33) is maximized to give the optimal thresholds. The results for 3-bit to 7-bit are listed in Table 5.1.

| Quantization bits | $s_1$ | $\Delta s$ |
|---|---|---|
| 3-bit | -69.14 | 5.005 |
| 4-bit | -71.76 | 2.848 |
| 5-bit | -72.94 | 1.42 |
| 6-bit | -73.6 | 0.85 |
| 7-bit | -73.9 | 0.435 |

Table 5.1: Estimated threshold values for 3-bit to 7-bit quantization.

### 5.5.2 Simulations of the centralized algorithm

In this subsection, we compare the ML-PSO method in section 5.2 with the CRLB of QRSS based localization as shown Section 2.3.3. The number of the particles is 30 and the maximum iteration is 100 in the centralized simulations.

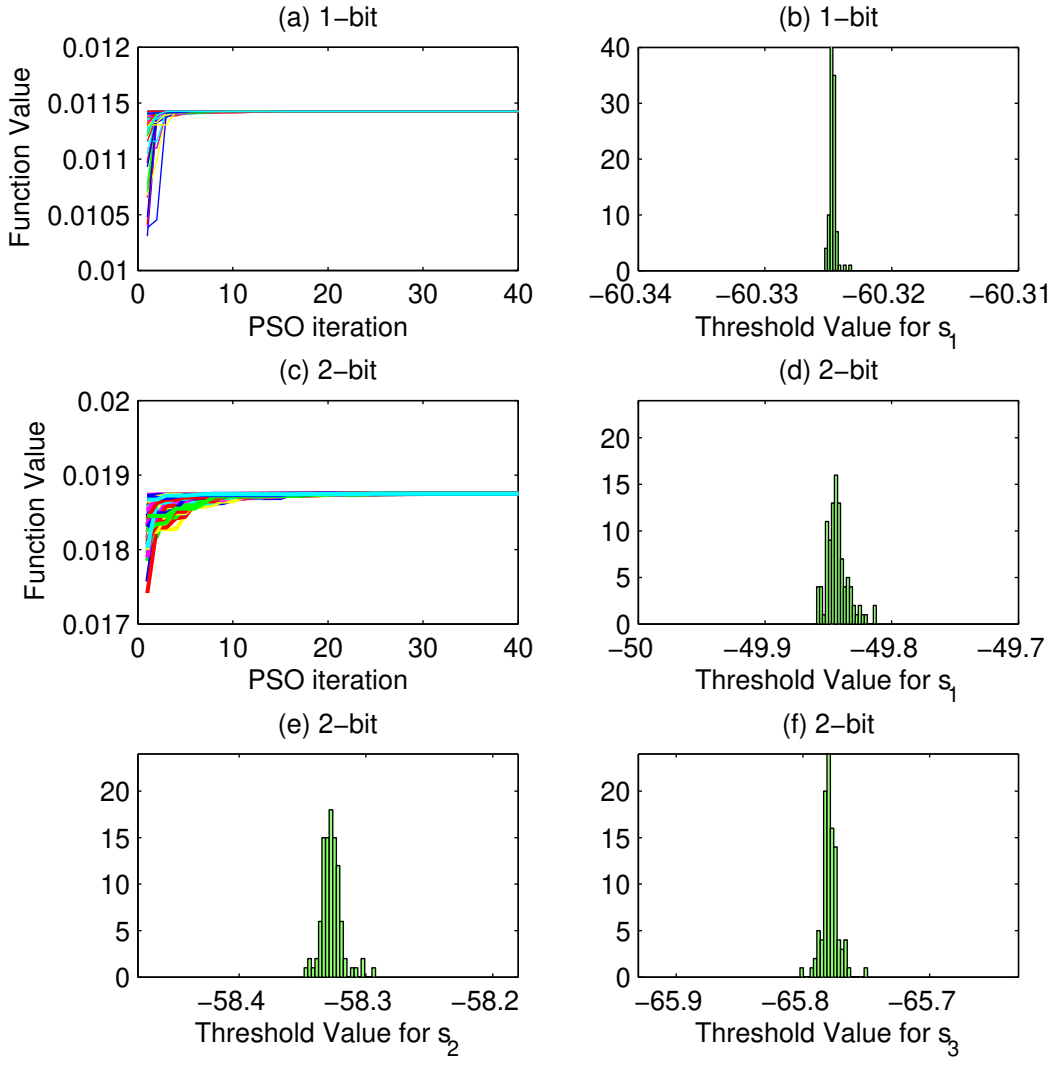



Figure 5.4: The PSO performance for threshold design. (a) and (c) are the cost function value convergence over 100 Monte Carlo runs for 1-bit and 2-bit quantization respectively. (b) is the histogram of the threshold value for 1-bit quantization. (d), (e), (f) are the histogram of the thresholds for 2-bit quantization.

A 100m by 100m surveillance area was considered for the simulations. A target is placed at $[40, 50]$m. The nodes in the network are assumed to be deployed similarly to the configuration shown in Fig. 5.1. Notice that the threshold design method has no relationship with the target and nodes positions. We assume that all nodes employ identical threshold values and the shadow noise $\sigma$ is 6dB.

In Fig. 5.5, the RMSE of the proposed estimator using binary data (1 bit) and quaternary data (2 bits) is compared with corresponding CRLB and the NLS method. For each scenario, 600 Monte Carlo simulations are performed. The root mean square error (RMSE) is plotted as a function of the number of nodes $N$.

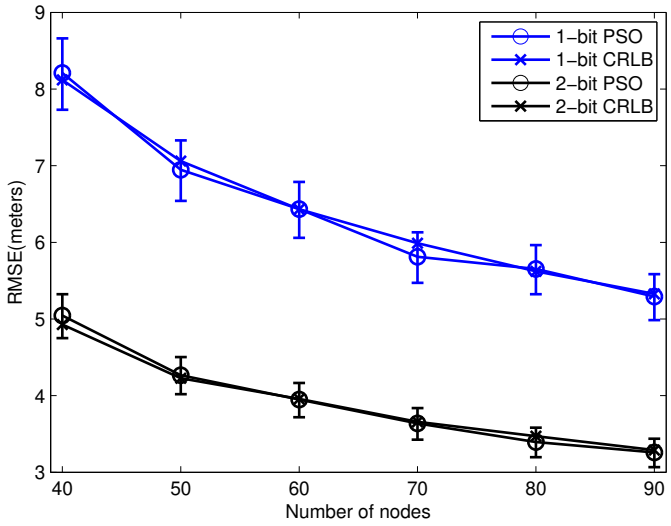



Figure 5.5: RMSE of PSO estimator and QRSS-CRLB. Quantization threshold for binary data is $s = -60.32$dBm. Quantization thresholds for quaternary data is $\mathbf{s} = [-49.85, -58.33, -65.78]$dBm.

As mentioned in Section 2.3, while using binary data, the QRSS data are known as proximity or connectivity measurements. If the RSS measurement is above the threshold, the node and target is in the range of communication. Therefore, the node transmits 1 to the fusion centre. If the RSS measurement is below the threshold, the node and target is out of the communication range, hence 0 is transmit to the fusion centre. Apparently, with proper quantization threshold, if there are both a large number of nodes transmitting 1s (in range) and 0s (out of range), the position of target will be well located. Therefore, as shown in Fig. 5.5, when the number of nodes $N$ increases, both the performance of ML-PSO estimator and the NLS estimator increase significantly. Similar results can be seen in Fig. 5.5 when quaternary data is employed.

In Fig. 5.6, the performance of the proposed estimator is evaluated employing different

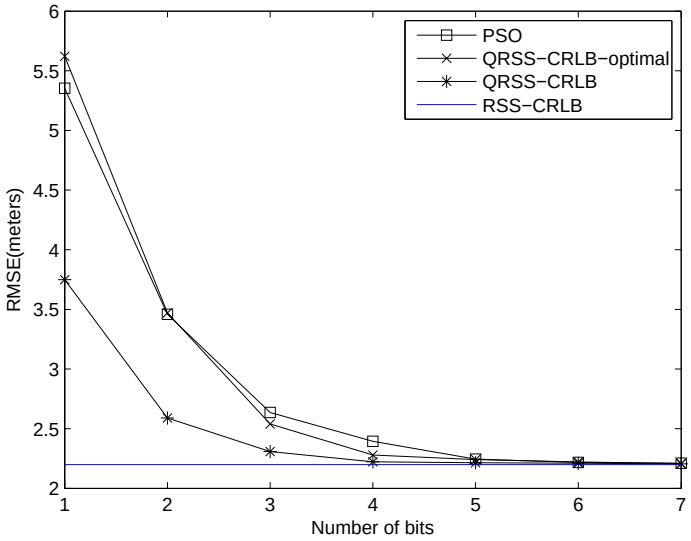



Figure 5.6: RMSE for PSO estimator using quantized data.

quantization bits. The performance of optimal threshold is added for comparison. The optimal threshold is obtained by minimizing the right hand side of (2.17) using PSO. Again, the simulation is based on 600 independent realizations. As it can be seen, in Fig. 5.6 the PSO estimator using quantized RSS data is very close to its CRLB. As number of quantization bit increases, the QRSS-CRLB improves and converges to the RSS-CRLB. Using only 5 bits, the QRSS-CRLB has little difference from the RSS-CRLB. Also, it can be clearly seen that the performance of PSO can attain the QRSS-CRLB at all scenarios. With the optimal threshold, the QRSS-CRLB is closer to the RSS-CRLB, especially at low quantization bit, because the optimal thresholds are only designed for this certain configuration.

### 5.5.3 Simulations of the distributed algorithm

In the next simulations, we evaluate the performance of the distributed algorithm proposed in Section 5.4. In order to study the performance of distributed algorithm using different number of quantization bits in the data exchange phase, we reduce the quantization error in the RSS measurement by using 7-bit QRSS data in the following simulations.

At first, the network is initialized such that each node is aware of its adjacent nodes' positions and their 7-bit QRSS measurement, hence each node can determine their metropolis weight and have an initial estimate using the PSO estimator. Then the initial estimate is quantized using (5.20) for further distributed processing. As we have no prior information of the target location, the

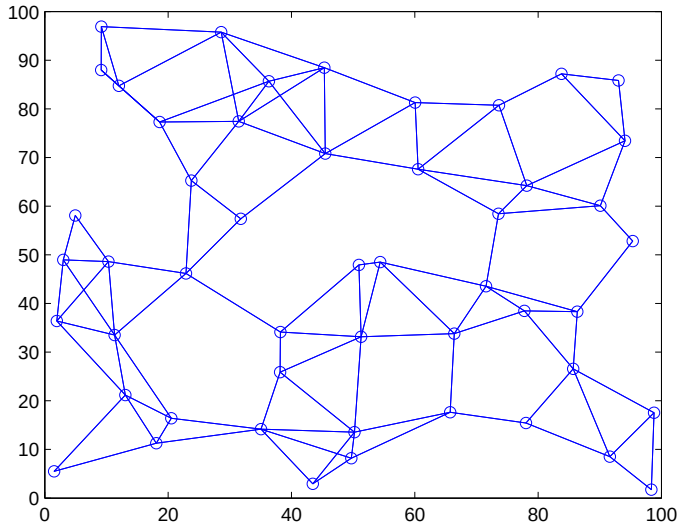



Figure 5.7: Network topology with 50 nodes.

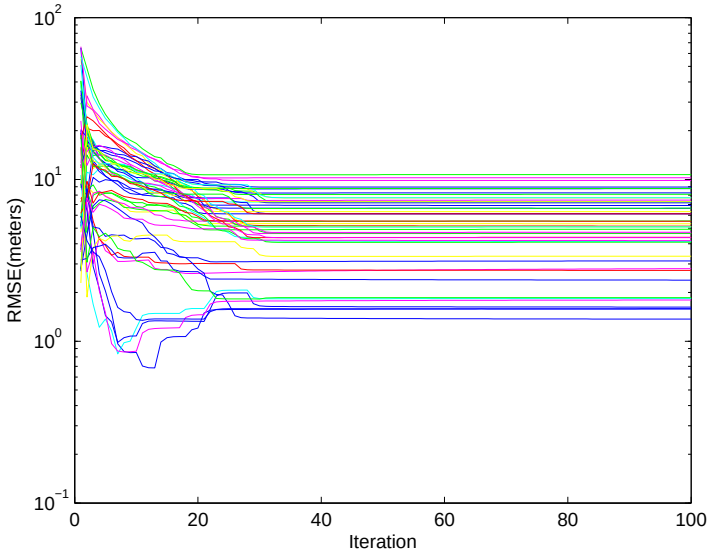



Figure 5.8: RMSE convergence of local estimates on 50 nodes in the distributed algorithms for QDG-I.

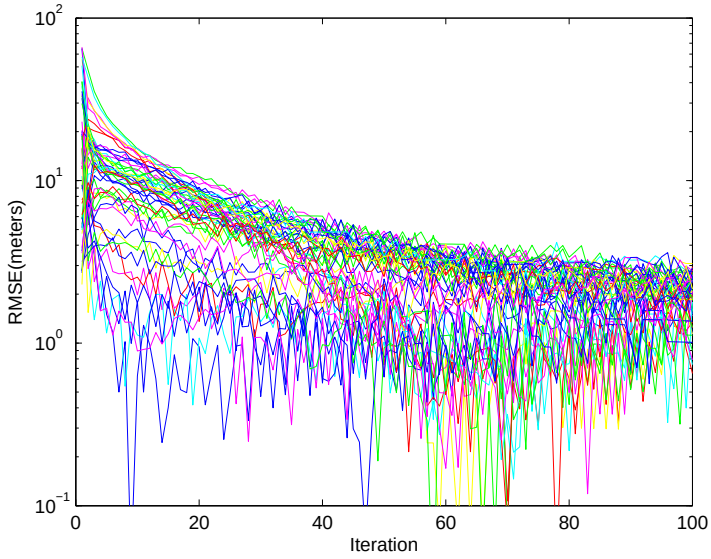



Figure 5.9: RMSE convergence of local estimates on 50 nodes in the distributed algorithms for QDG-II.

quantization range for the simulations is the range of surveillance area. Note that, in reality, if we have prior information of the target location, we could narrow the quantization range, such that the same performance can be achieved using less quantization bits in (5.20). In this example, we consider a network containing 50 nodes deployed over an area of 100m by 100m. The target is located at [40, 50]m. The communication range $d_c$ is 20m for each node. The network topology is shown in Fig. 5.7. The PSO is generally costly to completely solve an optimization problem. However, initialization does not need high accuracy. In this work, 20 particles and 40 iterations were used for initialization. The diminishing step size in both QDG-I and QDG-II is $\beta = 1/\sqrt{t}$ unless stated otherwise. The ARMSE is defined in (4.8).

Fig. 5.8 and Fig. 5.9 show how QDG-I and QDG-II converges within 100 iterations. Each line represents the RMSE of the local estimate at each node in the iteration process. Both algorithms are given the same initializations. It is also possible to notice that, after certain consensus iterations using QDG-I, the estimate at each node become inactive. However, in QDG-II, every node is still active and converging to better values since the consensus part in QDG-II can reduce part of quantization error.

Fig. 5.10 shows the average RMSE of the location estimate for QDG-I and QDG-II using different number of quantization bits versus the variance of shadowing. For comparison purposes, the distributed gradient algorithm using RSS measurements (DG-RSS), DPSO, RSS-CRLB and QRSS-CRLB are also included. The DG-RSS algorithm is computed using (5.21) with RSS

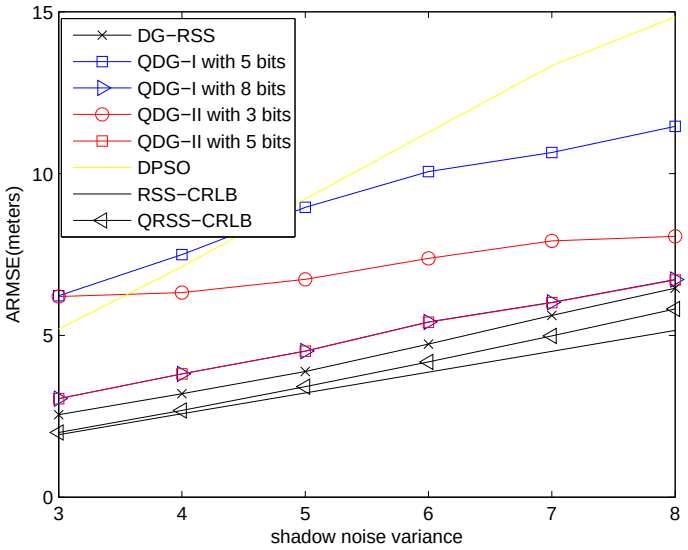



Figure 5.10: RMSE versus variance of shadowing noise plots in location estimation using different algorithms.

measurements and unquantized communication. The DPSO also uses RSS measurements and unquantized communication. The number of particles at each sensor node is 20 and the maximum iteration is 100. All simulations in Fig. 5.10 includes 600 Monte Carlo simulations.

As expected, the two CRLBs are very close to each other since 7-bit QRSS data is used. QDG-I with 8 bits has similar performance compared with the DG algorithm using RSS measurements at different level of shadowing noise, because the quantization level is high enough. The performance of QDG-II with 5 bits is almost the same as the QDG-I with 8 bits since QDG-II uses a quantization error reducing consensus algorithm. While 3 bit quantization is employed in QDG-II, the quantization noise is too large and has more influence on localization accuracy than shadowing noise. Therefore, it is shown in Fig. 5.10 that QDG-II with 3 bits has very large RMSE when the variance of shadowing is relatively low. Both QDG-I and QDG-II outperform DPSO especially when the noise is high.

The convergence behavior of the distributed algorithm at shadowing noise $\sigma = 5$ is depicted in Fig. 5.11. Each line represents the average RMSE over 300 independent trials in the iteration process. QDG-I with 5 bits converges fastest with worst performance. QDG-I with 8 bits has similar convergence rate with DG-RSS. Thus, to reach certain accuracy, we need to increase the quantization level. However, QDG-II with 5 bits also has similar convergence rate as QDG-I with 8 bits which means at certain performance accuracy, QDG-II costs less communication than QDG-I.

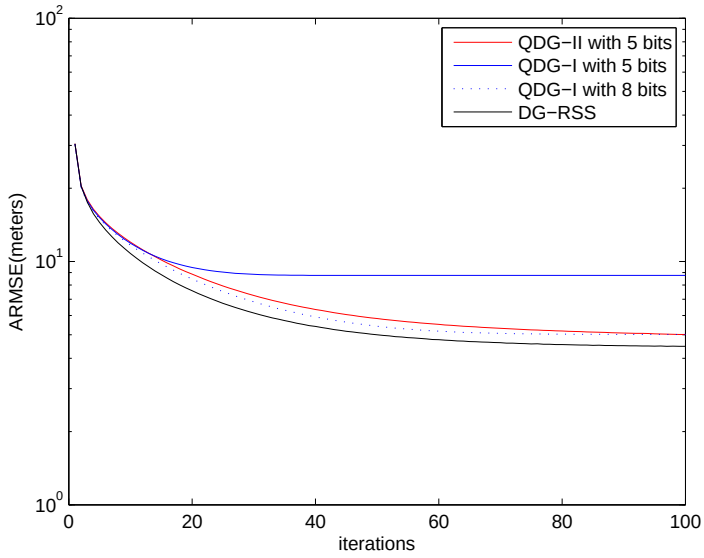



Figure 5.11: RMSE variation using QDG-I and QDG-II with different quantization bits.

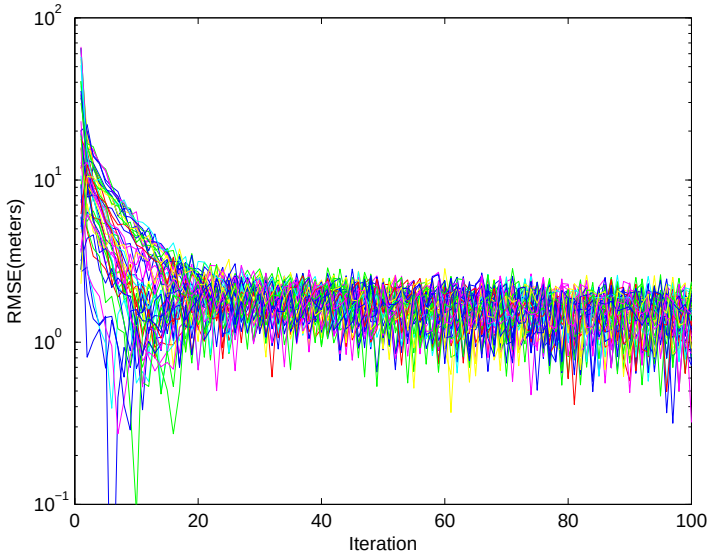



Figure 5.12: RMSE convergence of local estimates on 50 nodes using QDG-II.

In the next simulations, with the same scenario settings as shown in Fig. 5.7, we test the convergence of QDG-II algorithm with node communication range $d_c = 35m$.

In Fig. 5.12, we plot the localization RMSE convergence when nodes communication range is 35m. For comparison purpose, we use the same initialization as in Fig. 5.8. Compared with Fig. 5.8 and Fig. 5.9, the estimates of local nodes are converging much faster.

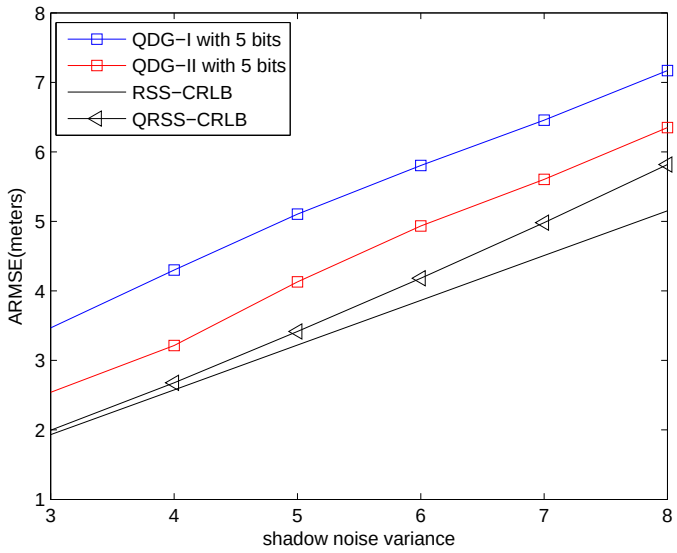



Figure 5.13: RMSE versus variance of shadowing noise plots in location estimation using QDG-I and QDG-II with 5 bits.

The same experiment in Fig. 5.10 is carried out in Fig. 5.13 while the communication range is 35m. Compared with the results in Fig. 5.10, the performance of both algorithms gets closer to CRLB, especially QDG-I.

It is important to test the performance of the distributed algorithms for various network configurations. Another network with 25 nodes shown in Fig. 5.14 is also simulated. The communication range $d_c$ in the network is 30m. Other system parameters are the same as previous simulation. The performance in terms of RMSE is shown in Fig. 5.15. Similarly, with the help of the quantization elimination schemes, QDG-II is close to the DG with unquantized data and outperforms QDG-I.

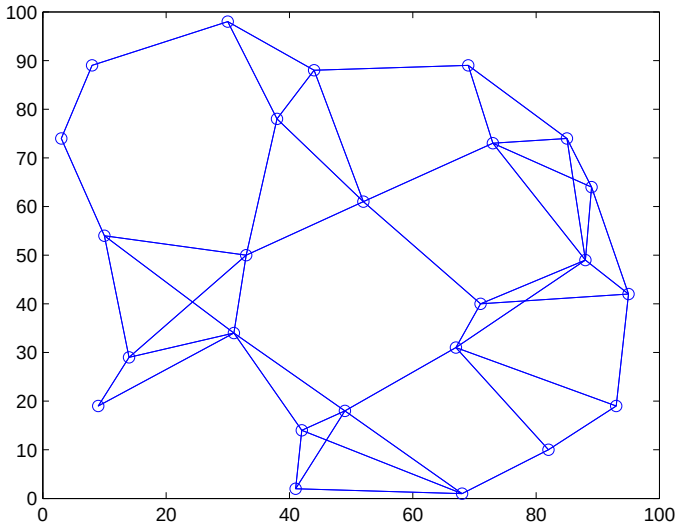



Figure 5.14: Network topology with 25 nodes.

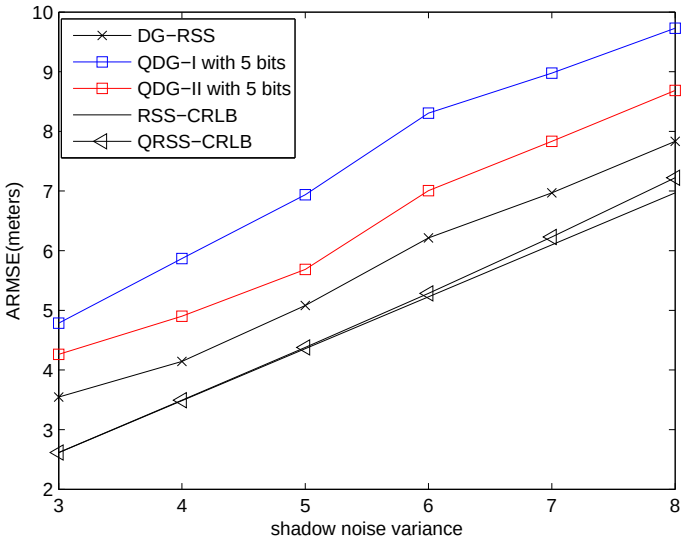



Figure 5.15: RMSE versus variance of shadowing noise plots in location estimation using QDG-I and QDG-II with 5 bits in a 25-node network.

## 5.6 Conclusion

In this chapter, we present a distributed target location estimation method that uses quantized data for WSNs based on a statistical RSS model. A practical threshold design method for RSS data is presented. Using this method, the CRLB of QRSS converges to the CRLB of RSS data as the quantization level increases. Then we solve the ML cost function of QRSS model using PSO. The PSO estimator can reach the CRLB with a relatively small number of quantization bits and enough number of nodes. Using the estimator described above, we initialize each node with local QRSS data using PSO and transmit quantized local estimates at each iteration. The quantized distributed gradient algorithm is applied to solve the ML cost function in a distributed manner. Simulation shows that with sufficient quantization levels in the local estimate, there is little difference between the distributed algorithm using quantized data and unquantized data. To reduce the number of bits used in the local estimate, a quantization compensation term is added into the distributed algorithm. Simulations show that with the usage of the compensation term, the QDG method has similar convergence rate and speed as the distributed gradient algorithm using unquantized data.

# Chapter 6

# Conclusions and Future Work

In this chapter, we summarise the contributions of this thesis for target localization using RSS measurements in WSNs. We draw some conclusions as well as indicating some avenues for future work.

## 6.1 Conclusions

The main problem investigated here was that of target localization using RSS measurements in the context of WSNs. The RSS-based localization problem was solved by considering correlated shadowing, unknown transmit power, and unknown PLE. The QRSS-based localization problem was tackled also with the help of PSO. Distributed algorithms were developed for further use in distributed WSNs.

In chapter 3, we studied the target localization problem using RSS measurements under correlated shadowing noise. In this work, the original unbiased estimate of distance proposed in [42] is extended to a correlated case. Inspired by the constrained total LS [67] proposed for TDoA based localization, the CWLS for RSS and DRSS based localization is proposed. The simulation results showed that CWLS outperforms most existing algorithms in terms of localization accuracy. Under the correlated noise assumption, localization accuracy of all algorithms improves, and CWLS has the best performance among all the algorithms. However, when DRSS measurement is recruited in correlated shadowing, the performance of CWLS only improves a little. Furthermore, CWLS is easy to implement, with low computation cost. CWLS achieves the CRLB in most scenarios.

In chapter 4, the PSO was used to optimise the RSS-based location estimation ML objective function. The use of PSO in the RSS-based localization has been studied previously in [73]. In [73], the author used a log-barrier method to improve the performance of the ML objective function when only four measurements are available. Here in chapter 4, the performance of PSO is tested without the log-barrier. The simulation results show that PSO has the ability to reach

the CRLB when the PLE is unknown. Unlike other existing methods, PSO does not require prior information on the PLE. Then, the DPSO proposed in [78] for dealing with bearing estimation problem is adapted to solve the distributed RSS based localization problem. The simulation results demonstrate that the performance of DPSO is acceptable in terms of RMSE. Even when there is a gap between the global and distributed estimate, DPSO has better performance than some existing methods. More importantly, it has the ability to locate the target when the PLE and transmit power are unknown.

In chapter 5, based on the QRSS model, we presented a distributed target location estimation method that uses quantised data for distributed WSNs. Following the work in [87], a practical threshold design method for QRSS data was presented. Using this method, the CRLB of QRSS converges to the CRLB of RSS data as the quantisation level increases. Then, we solved the ML cost function of the QRSS model using PSO. The simulations show that the PSO estimator can reach the CRLB of QRSS based localization.. To develop the quantised distributed algorithm, we initialised each node with local and adjacent QRSS measurements using PSO, and transmitting a quantised local estimate at each iteration. Then, the QDG algorithm [90] is employed to process the quantized data distributely. To improve the performance of QDG, a quantization compensation term is added to the original QDG. The simulation results show that with the usage of the compensation term, the QDG method has similar convergence rate and speed as the distributed gradient algorithm using unquantised data.

## 6.2 Suggestions for Future Work

Several aspects of RSS-based localization could be further developed.

1. One of the main advantages of the distributed positioning method is its extension to distributed tracking scenarios. The DPSO approach proposed in Chapter 4 could be combined with use of Kalman filter (KF) or unscented Kalman filter (UKF).

2. The threshold design method requires the transmit power of the target and the PLE of the environment. The threshold design method could be extended to cases of unknown transmit power or unknown PLE. An approach could be to assume that the transmit power or PLE follows some kind of distribution. The probability of received signal strength could then be calculated. Further, the same design method as that presented in Chapter 5 could be used to optimise the threshold.

3. An important application of QRSS is in cooperative node localization or multi-target localization, in which the targets transmit and receive signals among each other. The gradient or Gauss-Newton method can be used directly to solve the problem if initialisation is provided. However, in most scenarios, initialisation is not practical. For RSS-based localization methods, only Multidimensional Scale (MDS) and convex optimisation methods can solve this problem. However, as can be found in the literature, MDS may cause node ambiguity when the number of connections in the sensor network is low. Thus, the only methods that can retain the CRLB are convex optimisation methods, i.e. SDP and second order cone programming (SOCP). Recall the ML cost function of QRSS data model, the biggest challenge lies in how to relax the ML cost function into SDP or SOCP problems.

# Appendix A

# A Brief Introduction to Graph Theory

Graph theory concerns the relationship among lines and points. A graph is a nonempty finite set of vertices $\mathbf{V}$ along with a set $\mathbf{E}$ of 2-element subsets of $\mathbf{V}$. The element of $\mathbf{V}$ are called vertices, and the elements of $\mathbf{E}$ are called edges. An example of a simple graph is shown in Fig. A.1.

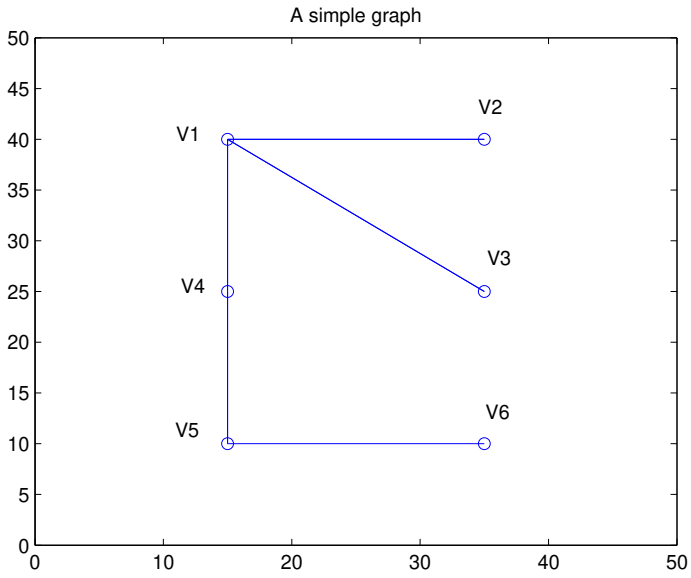



Figure A.1: An example.

A simple graph is an unweighted, undirected graph containing no graph loops or multiple edges. In the example above, the vertex set is $\mathbf{V} = \{V_1, V_2, V_3, V_4, V_5, V_6\}$, and the edge set is $\mathbf{E} = \{\{V_1, V_2\}, \{V_1, V_3\}, \{V_1, V_4\}, \{V_4, V_5\}, \{V_5, V_6\}\}$. The Graph above is also a connected graph. A graph is said to be connected when there is a path between every pair of vertices. A graph that is not connected is said to be disconnected. If every pair of distinct vertices of a graph is connected by a unique edge, the graph is said to be a complete graph or the graph is fully connected. The node-connectivity of a graph is equal to the minimum number of vertices that must be removed to disconnect the graph. In the example in Fig. A.1, the node-connectivity is 1 since with the removal of vertex $V_1$ leads to disconnection of the graph.

A simple graph can be represented using a square matrix called adjacency matrix. Two vertices of a graph are adjacent if there is an edge connecting them. The adjacency matrix of a graph is an $N * N$ symmetric binary matrix $\mathbf{A}_g$ with $[\mathbf{A}_g]_{ij} = 1$ if vertices $V_i$ and $V_j$ are connected by a unique edge, otherwise $[\mathbf{A}_g]_{ij} = 0$. The adjacency matrix in the example is

$$\mathbf{A}_g = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{A.1}$$

The degree of a vertex $V_i$ of a graph is the number of edges which are coming out from $V_i$. The degree matrix containing degree of all vertices is a diagonal matrix $\mathbf{D}_g = \text{diag}(d_{g_1}, \ldots, d_{g_N})$, where $d_{g_i}$ is the degree of vertices $V_i$. The degree matrix of the example is

$$\mathbf{D}_g = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{A.2}$$

The degree matrix can be used together with the adjacency matrix to construct the Laplacian matrix of a graph. The Laplacian matrix $\mathbf{L_p}$ is defined as

$$\mathbf{L_p} = \mathbf{D}_g - \mathbf{A}_g. \tag{A.3}$$

Note that the Laplacian matrix is positive semidefinite with eigenvalues that satisfy

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N. \tag{A.4}$$

The smallest eigenvalue of the Laplacian matrix is $0$, as the vector of all ones is an eigenvector of $\mathbf{L_p}$. The second smallest eigenvalue of $\mathbf{L_p}$ is called the algebraic connectivity. The algebraic connectivity is a lower bound on the node-connectivity. Its value is also related to the convergence speed of the consensus algorithm [98].

# Appendix B

# A Brief Introduction to Wireless Sensor Networks

This Appendix is mainly based on the work in [99], [100] and [101].

WSNs can be defined as a self-configured and infrastructureless wireless networks to monitor events and phenomena in a specified environment, such as temperature, sound, vibration, pressure, motion or pollutants and to pass their data through the network to a main location or base station where the data can be observed and analyzed. The environment can be the physical world, a biological system or an information technology framework. The base station acts like an interface between users and the WSNs. The user could be a civil, governmental, commercial or industrial entity. In most cases, a wireless sensor network is consists of hundreds or thousands of sensor nodes. The sensor nodes can communicate with each other within the network by wireless signal. This appendix includes a brief introduction of applications of WSNs, structure of the wireless sensor node and basic notion of network connectivity.

Wireless sensor networks have gained considerable popularity due to their flexibility in solving problems in different application domains and have the potential to change our lives in many different ways. WSNs have been successfully applied in various application domains, such as:

Military applications: WSNs can be used as an integral part of military communications, computing, control, battlefield surveillance and targeting systems.

Environmental applications: WSNs can cover many applications such as Forest fire detection, Flood detection, Precision agriculture, Landslide detection.

Commercial applications: Wireless sensor networks have been developed for Real-time vehicle tracking and detection. And it can also be utilized to monitor and control the Environmental in industrial and office buildings.

Health applications: Some of the health applications for sensor networks are supporting interfaces

for the disabled, integrated patient monitoring, diagnostics, and drug administration in hospitals.

A sensor node is made up of four basic components such as sensing unit, processing unit, transceiver unit and a power unit. It also has application dependent additional sensing units. Sensing units are usually composed of two subunits: sensors and analogue to digital converters (ADCs). The analogue signals collected by the sensors are converted to digital signals using ADC, and then passed over to the processing unit. The processing unit is generally associated with a small storage unit, typically flash memory. A transceiver unit connects the node to the network. The power units are typically batteries. With multiple units within one device, the nodes in a WSN are inherently constrained by limited processing speed, storage capacity, and communication bandwidth.

There are a lot of challenges in the designation of the WSNs, such as fault tolerance, scalability of the network, production costs, power consumption. One fundamental challenge in network design is the network connectivity. If the network is modeled as a graph with the nodes as vertices and the communication link between a pair of nodes as an edge, then a connected-network implies that the underlying graph is connected, i.e., between any two nodes there exists a single-hop or multi-hop communication path consisting of consecutive edges in the graph. Similar to the notion in graph theory, the communication graph of the WSNs can be denoted as $\mathbf{G}_c = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is the set of nodes and $\mathbf{E}$ is the set of edges such that an edge exists between any two nodes if their Euclidean distance is less than the communication radius. Based on the communication graph, the degree of a node is defined as the number of its adjacent nodes. The network formed by the induced communication graph is said to be connected if for every pair of nodes there exists a single-hop or a multi-hop path in $\mathbf{G}_c$, otherwise the network is said to be disconnected. The network is said to be $k$-node connected if for any pair of nodes there are at least $k$ mutually node-disjoint paths connecting them. The robustness of the network can be analyzed using the Laplacian matrix.

# Appendix C

# A Brief Overview of Optimization

In engineering design, optimization is to choose parameters to improve some objectives. Optimization involves the selection of the optimal solution from the set of candidate solutions. The quality of the solution is evaluated using an objective function (cost function) which is to be minimized or maximized. The search process is undertaken subject to the system model and restrictions which are termed constraints. These constraints are in the form of equality and/or inequality expressions.

A general optimization problem can be formulated as

$$\min_{\mathbf{x}\in\mathbb{R}^n} f_0(\mathbf{x}). \tag{C.1}$$

subject to $m$ constraints:

$$f_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \ldots, m, \tag{C.2}$$

where $\mathbf{x}$ is a vector contains the variables to be chosen, $f_0$ is the objective function to be minimized $f_1, \ldots, f_m$ are the inequality constraint functions. Note that maximizing $g(\mathbf{x})$ corresponds to $f_0(\mathbf{x}) = g(\mathbf{x})$, and the constraint could also be equality constraint. $\mathbf{x}$ represents the parameters in a model, constraints impose requirements on model parameters (e.g., nonnegativity), objective $f_0(\mathbf{x})$ is the evaluation on some observed data.

When it comes to solving the optimization problem, there are many important considerations, for instance global vs local optimization, convex vs non-convex optimization, gradient-based vs. derivative-free algorithms. Numerous optimisation algorithms have been developed over the past several decades. Each algorithm is usually restricted to special cases with strengths and weaknesses.

For general nonlinear functions, most algorithms only guarantee a local optimum. A more difficult problem is to find a global optimum: the optimum of $f_0$ for all feasible $\mathbf{x}$. The difficulty increases exponentially as the dimension of the problem $n$ increases. Many algorithms such as genetic algorithm (GA) and simulated annealing (SA) can be utilized to deal with this problem. However, it is very difficult to guarantee that a global minimum is found, since most algorithms may get stuck in local optimum.

One important subfield of optimization is convex optimization. Convex optimization studies the case when the objective function and all the constraint functions are convex. That is,

$$f_i(\beta_v \mathbf{x} + (1-\beta_v)\mathbf{y}) \leq \beta_v f_i(\mathbf{x}) + (1-\beta_v) f_i(\mathbf{y}), \quad \beta_v \in [0,1]. \tag{C.3}$$

For a convex optimization problem, any local optimum must be a global optimum. Until now, nearly complete theory (e.g. duality, optimality conditions) has been built for convex optimization [102]. Effective algorithms (e.g. subgradient projection, Interior-point methods) and modeling languages (e.g. CVX (Matlab) and CVXPY (Python)) are available for convex optimization.

Some classic problems such as linear programming (LP) and semidefinite programming (SDP) are all convex problems. In LP, the objective function and constraints are linear (affine), thus a LP problem is also a convex optimization problem. In SDP, the constraints are positive semidefinite. A general form of SDP is expressed as:

$$\begin{aligned} \text{minimize} \quad & \mathbf{C} \cdot \mathbf{X} \\ \text{subject to} \quad & \mathbf{A}^i \cdot \mathbf{X} = \mathbf{b}^i, \quad i = 1, \ldots, n \\ & \mathbf{X} \succeq 0. \end{aligned} \tag{C.4}$$

Notice that in an SDP, the variable is the matrix $\mathbf{X}$. The objective function is the linear function $\mathbf{C} \cdot \mathbf{X} = \sum_i \sum_j [\mathbf{C}]_{ij} [\mathbf{X}]_{ij}$ and there are $n$ linear equations that $\mathbf{X}$ must satisfy, namely $\mathbf{A}_i \cdot \mathbf{X} = \mathbf{b}_i$. The variable $\mathbf{X}$ also is positive semidefinite.

# Appendix D

# A Brief Introduction to Distributed Optimisation

Due to increasing interest in distributed network systems, there is a need for the development of distributed algorithms to handle computation over the distributed network. Average consensus algorithms have the ability to compute the global average over the whole network using only local neighbouring data. By combining with consensus algorithms, distributed versions of Expectation Maximization (EM) [103], Kalman Filter (KF) [104], Convex optimization [31], [105] are proposed. The general procedure for distributed consensus-based optimization consists of two steps. In the first step, each node performs the optimisation of a local cost function using only local data. In the next step, nodes communicate with their adjacent nodes to share local estimates. Then, nodes update local estimates with estimates from neighbouring nodes.

In this section, we will review some basic concepts of distributed consensus-based optimization that will be used in Chapters 4 and 5.

As mentioned in Appendix B, the WSN can be modeled as an undirected graph $\mathbf{G}_c = (\mathbf{V}, \mathbf{E})$, whose vertices $\mathbf{V}$ represents the nodes, and whose edges $\mathbf{E}$ represents the available communication links among sensors. An edge between node $i$ and node $j$ exists if node $i$ can communicate with node $j$ directly. With $\mathbf{V}$ and $\mathbf{E}$, the Adjacency matrix $\mathbf{A}_g$ and Laplacian matrix $\mathbf{L_p}$ for the WSN can also be built. Let the vector of measurements on the nodes be

$$\mathbf{x} = [x^1, x^2, \ldots, x^N]^T, \tag{D.1}$$

where $x^i$ is the measurement taken by the $i$th node. The most common average consensus algorithm [106] is given by

$$x^i(t+1) = x^i(t) - \beta_c \sum_{j \in \mathcal{N}_i} (x^i(t) - x^j(t)), \tag{D.2}$$

where $\beta_c$ is the step size, $\mathcal{N}_i$ is the set of adjacent nodes of node $i$ and $t$ is the iteration step. Putting all $x^i$ together we have that

$$\mathbf{x}(t) = (\mathbf{I} - \beta \mathbf{L_p})\mathbf{x}(0) = \mathbf{W}^t \mathbf{x}(0), \tag{D.3}$$

where $\mathbf{x}(0)$ are the initial measures taken by the nodes. In order to ensure that the systems converges to the average of initials, $\mathbf{W}$ must satisfy

$$\lim_{t \to \infty} \mathbf{W}^t = \frac{1}{N} \mathbf{1}\mathbf{1}^T. \tag{D.4}$$

As shown in [107], there exist necessary and sufficient conditions for (D.4) to hold. The first condition is that $\mathbf{W}$ satisfies

$$\mathbf{W}\mathbf{1} = 1, \tag{D.5}$$

$$\mathbf{1}^T \mathbf{W} = 1. \tag{D.6}$$

The second condition is that

$$\varrho(\mathbf{W} - \frac{1}{N} \mathbf{1}\mathbf{1}^T) < 1, \tag{D.7}$$

where $\varrho(\mathbf{W})$ denotes the spectral radius of the matrix $\mathbf{W}$. The spectral radius of matrix $\mathbf{W}$ is defined as

$$\varrho(\mathbf{W}) = \max_i(|\lambda_i(\mathbf{A})|), \tag{D.8}$$

where $\lambda_i(\mathbf{W})$ denotes the $i$th eigenvalue of $\mathbf{W}$.

One of the most famous weighting matrix in consensus algorithms is the Metropolis Weight [97]. The Metropolis weight can preserve the average and is simple to compute. In particular, each node only needs to know the degrees of its neighbours to determine the weights on its adjacent edges. The Metropolis Weights is shown as follows:

$$\mathbf{W}_{ij} = \begin{cases} 1/(1+\max\{d_{gi}, d_{gj}\}) & \text{if } i \neq j \text{ are connected} \\ 1 - \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} & \text{if } i = j \\ 0 & \text{if } i \text{ and } j \text{ are not connected} \end{cases}, \tag{D.9}$$

where $d_{gi}$ is the number of adjacent nodes of node $i$ (degree of node $i$) and $\mathcal{N}_i$ denote the set of neighbours of node $i$. With average consensus algorithm, the nodes in a WSN can process the collected data distributedly. In the following, we introduce one simple distributed algorithm via consensus, the distributed gradient algorithm.

Consider that, in the WSN, $N$ nodes cooperatively minimize a common additive cost function, in particular, the unconstrained optimization problem can be expressed as

$$\underset{x}{\text{minimise}} \quad f(x) = \sum_{i=1}^{N} f_i(x) \tag{D.10}$$

where each $f_i$ is a convex function. If the involved functions $f_i(x)$ is convex, then $f(x)$ is convex, and the global minimizer can be found using an iterative descent algorithm like gradient method [92].

In the gradient algorithm, the system updates the current estimate $x(t)$ with previous estimate $x(t-1)$

$$x(t) = x(t-1) - \beta\zeta(t-1), \tag{D.11}$$

where $\beta > 0$ is the step size of the gradient and $\zeta(t)$ is the gradient of $f(x)$.

In the distributed gradient algorithm, by combining the gradient method with the consensus algorithm, node $i$ updates its estimate according to the following relation

$$x^i(t+1) = \sum_{j=1}^{N} \mathbf{W}_{ij} x^j(t) - \beta \zeta_i(t), \tag{D.12}$$

$\zeta_i(t)$ is the gradient of local cost function $f_i(x)$. Note that the first term on the right hand side of equation is the average consensus with weighting matrix $\mathbf{W}$.

It is proved in [108], that, as $t \to \infty$, the local estimate $x^i(t)$ gradually converges to the global estimate. There are also variants of the distributed gradient algorithm. In [109], a fast distributed gradient method is proposed. In this method, the estimate $x^i(t)$ is not the variable used in consensus. Instead, a modified auxiliary variable based on $x^i(t)$ is used in consensus. The results show that, with the use of auxiliary variables, the distributed gradient algorithm converges faster with higher accuracy compared with the original gradient method. A distributed Gradient method with constraint problems is studied in [110].

# Appendix E

# A Brief Introduction of Particle Swarm Optimisation

The PSO algorithm is one of the swarm intelligence (SI) algorithm, originally developed by Kennedy and Eberhart in 1995 [79]. It was motivated by the social behaviour of flocks of birds or schools of fish, and the technique shares many similarities with evolutionary computation techniques such as genetic algorithms (GA). In the most common implementations of PSO, particles move through the search space using a combination of an attraction to the best solution that they individually have found, and an attraction to the best solution that any particle in the swarm has found.

An individual particle $i$ is composed of three vectors: its position in the $D$-dimensional search space $\mathbf{u}_i = (u_{i_1}, u_{i_2}, \cdots, u_{i_D})$, the best position that it has individually found $\bar{\mathbf{p}}_i = (\bar{p}_{i_1}, \bar{p}_{i_2}, \cdots, \bar{p}_{i_D})$, and its velocity $\mathbf{v}_i = (v_{i_1}, v_{i_2}, \cdots, v_{i_D})$. PSO starts with particles in random positions, representing random guesses in the search space. These particles are candidate solutions to the problem under consideration. Each particle of the swarm is associated with a random velocity, which helps to update its position and velocity according to its own as well as the swarm's experiences, that is, the particles' best and the swarm's (global) best information. The velocity $\mathbf{v}_{i,k}(t+1)$ and position $\mathbf{u}_{i,k}(t+1)$ at $t$th iteration and $k$th dimension is updated as follows:

$$\mathbf{v}_{i,k}(t+1) = \mathbf{v}_{i,k}(t) + \varphi r_1(\bar{\mathbf{p}}_{i,k}(t) - \mathbf{u}_{i,k}(t)) + \varphi r_2(\mathbf{g}_k - \mathbf{u}_{i,k}(t)) \quad i = 1, \ldots, P, \tag{E.1}$$

$$\mathbf{u}_{i,k}(t+1) = \mathbf{u}_i(t) + \mathbf{v}_{i,k}(t+1), \tag{E.2}$$

where $r_1$ and $r_2$ are random value uniformly distributed in $[0 \quad 1]$. $\mathbf{p}_{i,k}^k(t)$ is the best position of particle $i$ at dimension $k$ has from time step 1 to $t$; and $\mathbf{g}_k$ is the swarm's best at current time step

$t$ of dimension $k$. $P$ is the number of particles in the swarm. $\varphi$ is a constant with value of 2.0. $r_1$ and $r_2$ are independent random numbers uniquely generated at every update for each individual dimension $k = 1, \cdots, D$.

In order to prevent the particles from travelling too fast, causing swarm divergence, a maximum velocity is used to restrict the particles. The velocity limit can be applied along each dimension at each node as follows:

$$\mathbf{v}_{i,k}(t+1) = \begin{cases} V_{k,\max} & \text{if } \mathbf{v}_{i,k}(t+1) > V_{k,\max} \\ -V_{k,\max} & \text{if } \mathbf{v}_{i,k}(t+1) < -V_{k,\max} \end{cases} \tag{E.3}$$

where $V_{k,\max}$ is the maximum velocity of particles at dimension $k$. Generally, $V_{k,\max}$ is limited to half of the search space range. If the new position of the particle exceeds the boundary of the search space, it is adjusted to lie within the search space. The iteration stops when the maximum iteration number $T_{\max}$ is reached.

Although the tests run by Kennedy and Eberhart [79] demonstrated that the original PSO performed well on some cost functions, its performance in most other widely used benchmark problems is far from satisfactory. In order to balance the local search and the global search during optimisation in the original PSO, the concept of an inertia weight [111], $w$, is introduced into (E.1), leading to the modified velocity update:

$$\mathbf{v}_{i,k}(t+1) = w(t)\mathbf{v}_{i,k}(t) + \varphi_1 r_1(\bar{\mathbf{p}}_{i,k}(t) - \mathbf{u}_{i,k}(t)) + \varphi_2 r_2(\mathbf{g}_k - \mathbf{u}_{i,k}(t)) \quad i = 1, \ldots, P_s. \tag{E.4}$$

By adjusting the value of $w$, the swarm has a greater tendency to eventually constrict itself down to the area containing the best fitness and explore that area in detail. The authors also suggested using $w$ as a dynamic value over the optimization process. According to the experiments reported in [112], when $w$ is linearly decreasing over iteration, the performance of PSO significantly improves. The time-decreasing strategy for $w(t)$ can be written as

$$w(t) = w_{\max} - \frac{(w_{\max} - w_{\min})}{T_{\max}}(t-1). \tag{E.5}$$

where $w(t)$ is a function of iteration step, and $w(t)$ is the inertial weight at $t$th iteration. $w_{\max}$

and $w_{\min}$ are the initial and final values of the inertia weight, respectively, $t$ is the current iteration number, and $T_{max}$ is the maximum number of iterations. Using the linearly decreasing strategy, at the beginning of the search within a run, PSO with large inertia weight can provide high diversity, enabling use of the full range of the search space, at the end of the search, a small inertia weight can improve convergence with the optimal solution. Simulations [112] show that with the linearly decreasing inertia weight in the range [0.9 0.4], the PSO algorithm will converge efficiently. In all simulations of the thesis, the PSO with inertial weight is referred to as PSO for notation simplicity. The algorithm for PSO is summarized as follows:

**Algorithm E.0.1:** The PSO Algorithm

• Define PSO parameters and problem search space.
• Randomly initialise particles' positions and velocities.
**for** *each iteration* $t$ **do**
  **for** *each particle* $i$ **do**
    Evaluate the cost function of current iteration for each particles;
    Update particle best location $\bar{\mathbf{p}}_i(t)$ and group best position $\mathbf{g}(t)$ according to fitness value;
    Update particle velocity according to (E.4);
    **if** *velocity exceeds maximum* **then**
      Clamp particle velocity according to (E.3);
    **end**
    Update particle position using (E.2);
    **if** *particle position out of boundary* **then**
      set random location;
    **end**
  **end**
  Check if $t$ reaches the maximum iteration number;
**end**

An example of the progressive behaviour of PSO is illustrated in Fig.E.1 with inertia weight in the range $[0.9 \quad 0.4]$. In this illustration, the PSO is employed to optimise the objective function of a location estimation problem. As shown in Fig.E.1(a), in the beginning, the particles are initialized at random positions in the search space. Fig.E.1(b) shows that at the 15th PSO iteration the particles have largely concentrated towards the target, which lies at [12, 7] m. In the end, as shown in Fig.E.1(c), all the particles arrive at the target, and the optimisation is completed.

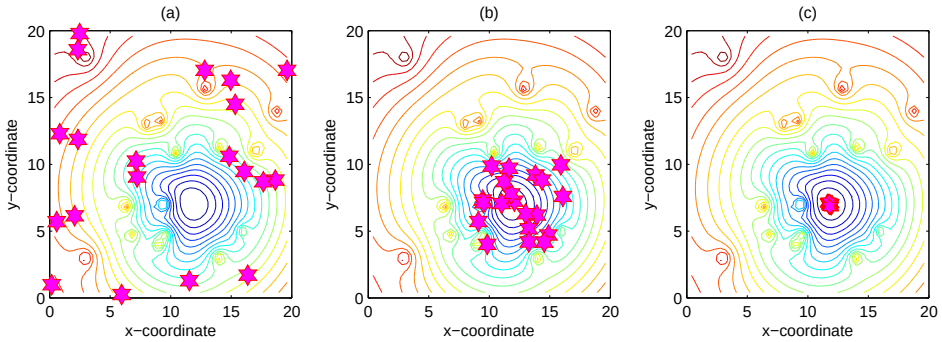



Figure E.1: An example of PSO progressive behaviour. The red hexagram represents 20 particles. (a), (b) and (c) show the particle position at the iterations $t = 1$, $t = 15$, and $t = 30$, respectively.

# Appendix F

# Performance Analysis of The CWLS

The performance of CWLS is analysed in this appendix. This work is an adaptation of the performance analysis of the constrained total LS for TDOA based localization [67]. For simplicity, the correlation coefficient $\rho$ is assumed to be zero.

The RSS measurement is contaminated by noise $n$, which can be viewed as a small perturbation. To understand how the random perturbation acting on the matrix $\hat{\mathbf{b}}$ propagates to the random perturbation $\Delta\mathbf{u} = \hat{\mathbf{u}} - \mathbf{u}$, we take partial derivatives of (3.27) with respect to $\mathbf{u}$. The gradient is (3.31). The solution $\hat{\mathbf{u}}$ minimize (3.31), however it is calculated, must be a zero of the gradient. Thus,

$$\frac{\partial J(\mathbf{u})}{\partial \mathbf{u}} \mid_{\mathbf{u}=\hat{\mathbf{u}}} = 2\boldsymbol{\gamma}_r^T \mathbf{Q}_r^{-1} \boldsymbol{\Gamma}_r = 0. \tag{F.1}$$

where

$$\boldsymbol{\gamma}_r(\hat{\mathbf{u}}) = \overline{\mathbf{A}}(\mathbf{u} + \Delta\mathbf{u}) + \mathbf{1}_N(\mathbf{u} + \Delta\mathbf{u})^T(\mathbf{u} + \Delta\mathbf{u}) - (\mathbf{b} + \Delta\mathbf{b}), \tag{F.2}$$

$$\boldsymbol{\Gamma}_r(\hat{\mathbf{u}}) = \overline{\mathbf{A}} + 2((\mathbf{u} + \Delta\mathbf{u})\mathbf{1}_N^T)^T. \tag{F.3}$$

Omit the second order perturbation $\Delta\mathbf{u}^T\Delta\mathbf{u}$ in (F.2), (F.2) can be approximated as

$$\boldsymbol{\gamma}_r(\hat{\mathbf{u}}) \approx \overline{\mathbf{A}}\Delta\mathbf{u} + 2\mathbf{1}_N(\mathbf{u}^T\Delta\mathbf{u}) - \Delta\mathbf{b}. \tag{F.4}$$

Substituting (F.4) and (F.3) into (F.1) gives

$$(\boldsymbol{\Gamma}_r\Delta\mathbf{u} - \Delta\mathbf{b})^T\mathbf{Q}_r^{-1}(\boldsymbol{\Gamma}_r + 2\mathbf{1}_N\Delta\mathbf{u}^T) = \mathbf{0}. \tag{F.5}$$

Note that $\mathbf{\Gamma}_r = \overline{\mathbf{A}} + 2(\mathbf{u}\mathbf{1}_N^T)^T$. Omit the second order perturbation in (F.5) gives:

$$(\mathbf{\Gamma}_r \Delta\mathbf{u} - \Delta\mathbf{b})^T \mathbf{Q}_r^{-1} \mathbf{\Gamma}_r = \mathbf{0}. \tag{F.6}$$

Thus,

$$\Delta\mathbf{u} = (\mathbf{\Gamma}_r^T \mathbf{Q}_r^{-1} \mathbf{\Gamma}_r)^{-1} \mathbf{\Gamma}_r^T \mathbf{Q}_r^{-1} \Delta\mathbf{b}. \tag{F.7}$$

As $\hat{\mathbf{b}}$ is unbiased, hence, $\Delta\mathbf{u}$ is unbiased. The covariance of $\Delta\mathbf{u}$ is given by

$$E[\Delta\mathbf{u}\Delta\mathbf{u}^T] = (\mathbf{\Gamma}_r^T \mathbf{Q}_r^{-1} \mathbf{\Gamma}_r)^{-1} \mathbf{\Gamma}_r^T \mathbf{Q}_r^{-1} E[\Delta\mathbf{b}\Delta\mathbf{b}^T] \mathbf{Q}_r^{-1} \mathbf{\Gamma}_r (\mathbf{\Gamma}_r^T \mathbf{Q}_r^{-1} \mathbf{\Gamma}_r)^{-1}. \tag{F.8}$$

Since the elements in $\Delta\mathbf{b}$ are zero-mean white random processes with covariance $\mathbf{Q}_r$, substituting $E[\Delta\mathbf{b}\Delta\mathbf{b}^T] = \mathbf{Q}_r$ into (F.8) produces:

$$E[\Delta\mathbf{u}\Delta\mathbf{u}^T] = (\mathbf{\Gamma}_r^T \mathbf{Q}_r^{-1} \mathbf{\Gamma}_r)^{-1}. \tag{F.9}$$

For sufficiently small noise conditions, using the Taylor Series expansion, such that

$$\exp(\frac{0.04 \ln^2(10) \sigma_i^2}{\alpha^2}) \approx 1 + \frac{0.04 \ln^2(10) \sigma_i^2}{\alpha^2}. \tag{F.10}$$

Substituting (F.10) into $\mathbf{Q}_r$, then $\mathbf{Q}_r$ is approximated as:

$$[\mathbf{Q}_r]_{ii} \approx d_i^4 (\frac{0.04(\ln 10)^2 \sigma_i^2}{\alpha^2}). \tag{F.11}$$

Note that

$$\begin{aligned}\boldsymbol{\Gamma}_r &= \overline{\mathbf{A}} + 2(\mathbf{u}\mathbf{1}_N^T)^T \\ &= \begin{bmatrix} 2x - 2x_1 & 2y - 2y_1 \\ \vdots & \vdots \\ 2x - 2x_N & 2y - 2y_N \end{bmatrix}\end{aligned} \tag{F.12}$$

and substituting (F.12) and (F.11) into (F.9) produces

$$E[\Delta\mathbf{u}\Delta\mathbf{u}^T] = \frac{(\ln 10)^2}{100\alpha^2} \begin{bmatrix} \sum_{i=1}^N \frac{(x-x_i)^2}{\sigma_i^2 d_i^4} & \sum_{i=1}^N \frac{(x-x_i)(y-y_i)}{\sigma_i^2 d_i^4} \\ \sum_{i=1}^N \frac{(x-x_i)(y-y_i)}{\sigma_i^2 d_i^4} & \sum_{i=1}^N \frac{(y-y_i)^2}{\sigma_i^2 d_i^4} \end{bmatrix}^{-1} \tag{F.13}$$

One can observe that (F.13) is identical to the CRLB of the RSS model (2.19). Note that, to prove the identity between the covariance and CRLB, several approximation are applied. Only when the noise $n_i$ is very small, the CWLS is unbiased and its localization accuracy is the same as CRLB.

The bias and covariance analysis of CWLS for DRSS based localization is very similar to the process above considering the similarity between the RSS model and the DRSS model. The procedure is summarised below.

Similar to the procedure from (F.1) to (F.4), for the DRSS case, we have:

$$\frac{\partial J(\mathbf{u})}{\partial \mathbf{u}} \mid_{\mathbf{u}=\hat{\mathbf{u}}} = 2\boldsymbol{\gamma}_d^T \mathbf{Q}_d^{-1} \boldsymbol{\Gamma}_d = 0. \tag{F.14}$$

where

$$\boldsymbol{\Gamma}_d(\hat{\mathbf{u}}) = \overline{\mathbf{G}}_{12} + 2\overline{\mathbf{G}}_3^o(\mathbf{u}^T - \mathbf{c}_1^T) + 2\overline{\mathbf{G}}_3^o \Delta\mathbf{u}^T + 2\Delta\mathbf{r}\mathbf{u}^T + 2\Delta\mathbf{r}\mathbf{c}_1^T. \tag{F.15}$$

$$\boldsymbol{\gamma}_d(\hat{\mathbf{u}}) \approx (\overline{\mathbf{G}}_{12} + 2\overline{\mathbf{G}}_3^o(\mathbf{u}^T - \mathbf{c}_1^T))\Delta\mathbf{u} + \Delta\mathbf{r}(\mathbf{u}^T\mathbf{u} + \mathbf{c}_1^T\mathbf{c}_1 - 2\mathbf{c}_1^T\mathbf{u}). \tag{F.16}$$

and $\overline{\mathbf{G}}_3^o$ is $\overline{\mathbf{G}}_3$ without the error $\Delta\mathbf{r}$.

Let $\boldsymbol{\Gamma}_d = \overline{\mathbf{G}}_{12} + 2\overline{\mathbf{G}}_3^o(\mathbf{u}^T - \mathbf{c}_1^T)$. Substituting (F.16) and (F.15) into (F.14) and omitting the second

order perturbations gives

$$(\mathbf{\Gamma}_d \Delta\mathbf{u} - \Delta\mathbf{r}(\mathbf{u}^T\mathbf{u} + \mathbf{c}_1^T\mathbf{c}_1 - 2\mathbf{c}_1^T\mathbf{u}))^T \mathbf{Q}_d^{-1}\mathbf{\Gamma}_d = \mathbf{0}. \tag{F.17}$$

Notice that $\mathbf{u}^T\mathbf{u} + \mathbf{c}_1^T\mathbf{c}_1 - 2\mathbf{c}_1^T\mathbf{u} = d_1^2$, thus,

$$\Delta\mathbf{u} = -d_1^2(\mathbf{\Gamma}_d^T\mathbf{Q}_d^{-1}\mathbf{\Gamma}_d)^{-1}\mathbf{\Gamma}_d^T\mathbf{Q}_d^{-1}\Delta\mathbf{r}. \tag{F.18}$$

As $\Delta\mathbf{r}$ is unbiased, hence, $\Delta\mathbf{u}$ is unbiased. Similar to (F.9),

$$E[\Delta\mathbf{u}\Delta\mathbf{u}^T] = d_1^4(\mathbf{\Gamma}_d^T\mathbf{Q}_d^{-1}\mathbf{\Gamma}_d)^{-1}. \tag{F.19}$$

Following (F.10) and (F.11), $\mathbf{Q}_d$ is approximated as:

$$\mathbf{Q}_d \approx \frac{4(\ln 10)^2}{100\alpha^2}\mathbf{R}\mathbf{Q}_v\mathbf{R}, \tag{F.20}$$

where $\mathbf{R}$ is a diagonal matrix:

$$\mathbf{R} = \begin{bmatrix} \frac{d_2^2}{d_1^2} & & \\ & \ddots & \\ & & \frac{d_N^2}{d_1^2} \end{bmatrix}. \tag{F.21}$$

Therefore, (F.19) can be further expressed as

$$E[\Delta\mathbf{u}\Delta\mathbf{u}^T] = d_1^4(\mathbf{\Gamma}_d^T\mathbf{R}^{-1}\mathbf{Q}_v^{-1}\mathbf{R}^{-1}\mathbf{\Gamma}_d)^{-1}, \tag{F.22}$$

Note that $d_1^{-2}\mathbf{\Gamma}_d^T\mathbf{R}^{-1} = \mathbf{\Omega}$ in (2.57), thus the covariance matrix of $\hat{\mathbf{u}}$ can be approximated to the CRLB of DRSS based localization.

# Appendix G

# Derivation of (5.10)

The derivation of (5.10) is shown in this Appendix.

As shown in (5.6) and (5.9), the pdf for $v_i$ is

$$f_v(v_i) = \frac{1}{\gamma}\begin{cases} \frac{\pi}{b^2} + \frac{v_i}{b^4} - \frac{4\sqrt{v_i}}{b^3} & 0 < v_i \le b^2 \\ \frac{2}{b^2}\arcsin(\frac{2b^2 - v_i}{v_i}) - \frac{v_i}{b^4} + \frac{4\sqrt{v_i - b^2}}{b^3} - 2/b^2 & b^2 < v_i \le 2b^2 \\ 0 & \text{otherwise} \end{cases}. \tag{G.1}$$

There is a single solution to the equation $z_i = P_0 - 5\alpha \log_{10} v_i$, given by $v_i = 10^{\frac{P_0 - z_i}{5\alpha}}$. For notation simplicity, let $g_i = 10^{\frac{P_0 - z_i}{5\alpha}}$. Hence, $|z_i'(v_i)| = \frac{5\alpha}{g_i \ln 10}$. Therefore,

$$f_z(z_i) = \frac{f_v(10^{\frac{P_0 - z_i}{5\alpha}})}{|z_i'(v_i)|}. \tag{G.2}$$

When $0 < v_i \le b^2$,

$$f_z(z_i) = \frac{\ln 10}{5\alpha} g_i \gamma (\frac{\pi}{b^2} + \frac{g_i}{b^4} - \frac{4\sqrt{g_i}}{b^3}) \quad \phi_1 < z_i \le P_0, \tag{G.3}$$

where $\phi_1 = P_0 - 5\alpha \log_{10} b^2$.

When $b^2 < v_i \le 2b^2$,

$$f_z(z_i) = \frac{\ln 10}{5\alpha} g_i \gamma (\frac{2}{b^2}\arcsin(\frac{2b^2 - g_i}{g_i}) - \frac{g_i}{b^4} + \frac{4\sqrt{g_i - b^2}}{b^3} - \frac{2}{b^2}) \quad \phi_2 < z_i \le \phi_1, \tag{G.4}$$

where $\phi_2 = P_0 - 5\alpha \log_{10} 2b^2$..

Combining the two equations above gives the final pdf in (5.10).

# Appendix H

# Derivation of Fisher Information (5.16)

The derivation of FIM on $z$ follows the same procedure to the derivation of FIM on the target location [32]. The details are given below.

By using (5.16), we have

$$\frac{\partial \ln p(K|z_i)}{\partial z_i} = \frac{1}{p(K|z_i)} \frac{\partial p(K|z_i)}{\partial z_i} \tag{H.1}$$

and

$$\frac{\partial^2 \ln p(K|z_i)}{\partial z_i^2} = -\frac{1}{p^2(K|z_i)} [\frac{\partial p(K|z_i)}{\partial z_i}]^2 + \frac{1}{p(K|z_i)} \frac{\partial^2 p(K|z_i)}{\partial z_i^2}. \tag{H.2}$$

Now, taking expectation of (H.2) with respect to $p(K|z_i)$

$$\begin{aligned} E[\frac{\partial^2 \ln p(K|z_i)}{\partial z_i^2}] &= \sum_{K=0}^{L-1} p(K|z_i) \{-\frac{1}{p^2(K|z_i)} [\frac{\partial p(K|z_i)}{\partial z_i}]^2 + \frac{1}{p(K|z_i)} \frac{\partial^2 p(K|z_i)}{\partial z_i^2}\} \\ &= \sum_{K=0}^{L-1} \{-\frac{1}{p(K|z_i)} [\frac{\partial p(K|z_i)}{\partial z_i}]^2 + \frac{\partial^2 p(K|z_i)}{\partial z_i^2}\}. \end{aligned} \tag{H.3}$$

Notice that $\sum_{K=0}^{L-1} p(K|z_i) = 1$, the second term of (H.3) is actually

$$\sum_{K=0}^{L-1} \frac{\partial^2 p(K|z_i)}{\partial z^2} = \frac{\partial^2}{\partial z_i^2} [\sum_{K=0}^{L-1} p(K|z_i)] = 0. \tag{H.4}$$

With

$$\frac{\partial Q((s_l - z_i)/\sigma)}{\partial z_i} = \frac{\exp(-\frac{(s_l - z_i)^2}{2\sigma^2})}{\sqrt{2\pi}\sigma}, \tag{H.5}$$

it is easy to show that

$$[\frac{\partial p(K|z_i)}{\partial z_i}]^2 = \frac{[\exp(-\frac{(s_l - z_i)^2}{2\sigma^2}) - \exp(-\frac{(s_{l+1} - z_i)^2}{2\sigma^2})]^2}{2\pi\sigma^2 p(K|z_i)}. \tag{H.6}$$

With (H.4) and (H.6), we can have

$$-E[\frac{\partial^2 \ln p(K|z_i)}{\partial z^2}] = \sum_{K=0}^{L-1} \frac{[\exp(-\frac{(s_l - z_i)^2}{2\sigma^2}) - \exp(-\frac{(s_{l+1} - z_i)^2}{2\sigma^2})]^2}{2\pi\sigma^2 p(K|z_i)}. \tag{H.7}$$

# Appendix I

# Derivation of The Gradient of QRSS Model

The Gradient of the local ML cost function (5.19) is derived as follows. From (5.19), we have

$$\begin{aligned}\frac{\partial(-f(K_i|\mathbf{u}))}{\partial x} &= -\frac{1}{p(K_i|\mathbf{u})}\frac{\partial p(K_i|\mathbf{u})}{\partial x} \\ &= -\frac{1}{p(K_i|\mathbf{u})}\Big(\frac{\partial Q((\mathbf{s}_l - P_0 + 10\alpha\log d_i)/\sigma)}{\partial x} \\ &\quad - \frac{\partial Q(\mathbf{s}_{l+1} - P_0 + 10\alpha\log d_i)/\sigma}{\partial x}\Big).\end{aligned} \tag{I.1}$$

With

$$\frac{\partial Q((\mathbf{s}_l - P_0 + 10\alpha\log d_i)/\sigma)}{\partial x} = \frac{10\alpha(x - x_i)\exp(\frac{(\mathbf{s}_l - P_0 + 10\alpha\log d_i)^2}{2\sigma^2})}{\sqrt{2\pi}\sigma\ln 10 d_i^2}, \tag{I.2}$$

the gradient is expressed as

$$\begin{aligned}&\frac{\partial(-\ln f(K_i|\mathbf{u}))}{\partial x} \\ &= -\frac{10\alpha(x - x_i)(\exp(\frac{(\mathbf{s}_l - P_0 + 10\alpha\log d_i)^2}{2\sigma^2}) - \exp(\frac{(\mathbf{s}_{l+1} - P_0 + 10\alpha\log d_i)^2}{2\sigma^2}))}{\sqrt{2\pi}\sigma\ln 10 d_i^2 p(K_i|\mathbf{u})}.\end{aligned} \tag{I.3}$$

# Bibliography


[1] Geoff Werner-Allen, Konrad Lorincz, Jeff Johnson, Jonathan Lees, and Matt Welsh. Fidelity and yield in a volcano monitoring sensor network. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, OSDI '06, pages 381–396, Berkeley, CA, USA, 2006. USENIX Association.

[2] Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, and Deborah Estrin. A Wireless Sensor Network For Structural Monitoring. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 13–24, New York, NY, USA, 2004. ACM.

[3] Branislav Kusy, Akos Ledeczi, and Xenofon Koutsoukos. Tracking mobile nodes using rf doppler shifts. In *ACM Conference on Embedded Networked Sensor Systems (SenSys*, 2007.

[4] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA '02, pages 88–97, New York, NY, USA, 2002. ACM.

[5] K. Khakpour and M. H. Shenassa. Industrial control using wireless sensor networks. In *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*, pages 1–5, April 2008.

[6] D. Basu, G. Moretti, G. Sen Gupta, and S. Marsland. Wireless sensor network based smart home: Sensor selection, deployment and monitoring. In *Sensors Applications Symposium (SAS), 2013 IEEE*, pages 49–54, Feb 2013.

[7] N. Patwari, A.O. Hero, M. Perkins, N.S. Correal, and R.J. O'Dea. Relative location estimation in wireless sensor networks. *Signal Processing, IEEE Transactions on*, 51(8):2137–2148, Aug 2003.

[8] Dragoş Niculescu and Badri Nath. DV Based Positioning in Ad Hoc Networks. *Telecommun. Syst.*, 22(1-4):267–280, January 2003.

[9] Dai Chen, Wei Wang, and Yong Zhou. Notice of Retraction An Improved DV-Hop Localization Algorithm in Wireless Sensor Networks. In *Computer and Communication Technologies in Agriculture Engineering (CCTAE), 2010 International Conference On*, volume 2, pages 266–269, June 2010.

[10] H. Chen, P. Huang, M. Martins, H. C. So, and K. Sezaki. Novel centroid localization algorithm for three-dimensional wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pages 1–4, Oct 2008.

[11] Hongbo Fan, Guanglin He, Siqian Tao, and Hongyu Xu. Weighted centroid localization algorithm based on improved RSSI ranging. In *Mechatronic Sciences, Electric Engineering and Computer (MEC), Proceedings 2013 International Conference on*, pages 544–547, Dec 2013.

[12] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, MobiCom '03, pages 81–95, New York, NY, USA, 2003. ACM.

[13] S. Gezici, Zhi Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE Signal Processing Magazine*, 22(4):70–84, July 2005.

[14] Toshihiro Mogi and Tomoaki Ohtsuki. TOA localization using RSS weight with path loss exponents estimation in NLOS environments. In *2008 14th Asia-Pacific Conference on Communications*, pages 1–5, Oct 2008.

[15] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero, R.L. Moses, and N.S. Correal. Locating the Nodes: Cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE*, 22(4):54–69, July 2005.

[16] Hing Cheung So and Lanxin Lin. Linear Least Squares Approach for Accurate Received Signal Strength Based Source Localization. *Signal Processing, IEEE Transactions on*, 59(8):4035–4040, Aug 2011.

[17] N. Salman, Y. J. Guo, A. H. Kemp, and M. Ghogho. Analysis of linear least square solution for RSS based localization. In *Communications and Information Technologies (ISCIT),*

*2012 International Symposium on*, pages 1051–1054, Oct 2012.

[18] R.W. Ouyang, A.K.-S. Wong, Chin-Tau Lea, and V.Y. Zhang. Received Signal Strength-Based Wireless Localization via Semidefinite Programming. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6, Nov 2009.

[19] S. Tomic, M. Beko, and R. Dinis. RSS-Based Localization in Wireless Sensor Networks Using Convex Relaxation: Noncooperative and Cooperative Schemes. *IEEE Transactions on Vehicular Technology*, 64(5):2037–2050, May 2015.

[20] T. Stoyanova, F. Kerasiotis, K. Efstathiou, and G. Papadopoulos. Modeling of the RSS Uncertainty for RSS-based Outdoor Localization and Tracking Applications in Wireless Sensor Networks. In *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, pages 45–50, July 2010.

[21] H. Noureddine, N. Gresset, D. Castelain, and R. Pyndiah. Auto-Regressive Modeling of the Shadowing for RSS Mobile Tracking. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5, June 2011.

[22] R. M. Vaghefi, M. R. Gholami, and E. G. Strom. RSS-based sensor localization with unknown transmit power. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 2480–2483, May 2011.

[23] R. M. Vaghefi, M. R. Gholami, R. M. Buehrer, and E. G. Strom. Cooperative Received Signal Strength-Based Sensor Localization With Unknown Transmit Powers. *IEEE Transactions on Signal Processing*, 61(6):1389–1403, March 2013.

[24] B. C. Liu and K. H. Lin. Distance Difference Error Correction by Least Square for Stationary Signal-Strength-Difference-Based Hyperbolic Location in Cellular Communications. *IEEE Transactions on Vehicular Technology*, 57(1):227–238, Jan 2008.

[25] Y. T. Chan, B. H. Lee, R. Inkol, and F. Chan. Received signal strength localization with an unknown path loss exponent. In *Electrical and Computer Engineering (CCECE), 2011 24th Canadian Conference on*, pages 000456–000459, May 2011.

[26] N. Salman, M. Ghogho, and A. H. Kemp. On the Joint Estimation of the RSS-based Location and Path-loss Exponent. *IEEE Wireless Communications Letters*, 1(1):34–37, February 2012.

[27] J. C. Liberti and T. S. Rappaport. Statistics of shadowing in indoor radio channels at 900 and 1900 MHz. In *Military Communications Conference, 1992. MILCOM '92, Conference Record. Communications - Fusing Command, Control and Intelligence., IEEE*, pages 1066–1070 vol.3, Oct 1992.

[28] S. S. Ghassemzadeh, R. Jana, C. W. Rice, W. Turin, and V. Tarokh. A Statistical Path Loss Model For In-Home UWB Channels. In *Ultra Wideband Systems and Technologies, 2002. Digest of Papers. 2002 IEEE Conference on*, pages 59–64, May 2002.

[29] J. H. Lee and R. M. Buehrer. Location Estimation Using Differential RSS with Spatially Correlated Shadowing. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6, Nov 2009.

[30] Benjamin Bejar and Santiago Zazo. A practical approach for outdoors distributed target localization in wireless sensor networks. *EURASIP Journal on Advances in Signal Processing*, 2012(1):1–11, 2012.

[31] J. Li, E. Elhamifar, I. J. Wang, and R. Vidal. Consensus with robustness to outliers via distributed optimization. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 2111–2117, Dec 2010.

[32] Neal Patwari and Alfred O. Hero, III. Using Proximity and Quantized RSS for Sensor Localization in Wireless Networks. In *Proceedings of the 2Nd ACM International Conference on Wireless Sensor Networks and Applications*, WSNA '03, pages 20–29, New York, NY, USA, 2003. ACM.

[33] Dinesh Rajan Jeff Kennington, Eli Olinick. *Wireless Network Design: Optimization Models and Solution Procedures*. Springer, 2010.

[34] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[35] Andreas F. Molisch. *Wireless Communications*. Wiley Publishing, 2nd edition, 2011.

[36] William C. Y. Lee. *Mobile communications engineering*. McGraw-Hill Professional Publishing, 1998.

[37] D. C. Cox, R. R. Murray, and A. W. Norris. 800-MHz Attenuation measured in and around suburban houses. *AT T Bell Laboratories Technical Journal*, 63(6):921–954, July 1984.

[38] R. Bernhardt. Macroscopic diversity in frequency reuse radio systems. *IEEE Journal on Selected Areas in Communications*, 5(5):862–870, Jun 1987.

[39] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[40] W. H. FOY. Position-Location Solutions by Taylor-Series Estimation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-12(2):187–194, March 1976.

[41] X. Zhou P. Deift. A Steepest Descent Method for Oscillatory Riemann–Hilbert Problems. asymptotics for the mkdv equation. *Annals of Mathematics*, 137(2):295–368, 1993.

[42] S. D. Chitte, S. Dasgupta, and Z. Ding. Distance Estimation From Received Signal Strength Under Log-Normal Shadowing: Bias and Variance. *IEEE Signal Processing Letters*, 16(3):216–218, March 2009.

[43] P. Tarrio, A. M. Bernardos, J. A. Besada, and J. R. Casar. A new positioning technique for RSS-based localization based on a weighted least squares estimator. In *Wireless Communication Systems. 2008. ISWCS '08. IEEE International Symposium on*, pages 633–637, Oct 2008.

[44] K. W. Cheung, H. C. So, W.-K. Ma, and Y. T. Chan. A Constrained Least Squares Approach to Mobile Positioning: Algorithms and Optimality. *EURASIP J. Appl. Signal Process.*, 2006:150–150, January 2006.

[45] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1, March 2014.

[46] R. M. Vaghefi and R. M. Buehrer. Received Signal Strength-Based Sensor Localization in Spatially Correlated Shadowing. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 4076–4080, May 2013.

[47] G. Wang and K. Yang. A New Approach to Sensor Node Localization Using RSS Measurements in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 10(5):1389–1395, May 2011.

[48] A. Beck, P. Stoica, and J. Li. Exact and Approximate Solutions of Source Localization Problems. *IEEE Transactions on Signal Processing*, 56(5):1770–1778, May 2008.

[49] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. A new method for the nonlinear

transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, Mar 2000.

[50] Y. T. Chan and K. C. Ho. A simple and efficient estimator for hyperbolic location. *IEEE Transactions on Signal Processing*, 42(8):1905–1915, Aug 1994.

[51] M. R. Gholami, R. M. Vaghefi, and E. G. Strom. RSS-Based Sensor Localization in the Presence of Unknown Channel Parameters. *IEEE Transactions on Signal Processing*, 61(15):3752–3759, Aug 2013.

[52] G. Wang, H. Chen, Y. Li, and M. Jin. On Received-Signal-Strength Based Localization with Unknown Transmit Power and Path Loss Exponent. *IEEE Wireless Communications Letters*, 1(5):536–539, October 2012.

[53] A. K. M. Mahtab Hossain and Wee-Seng Soh. Cramer-Rao Bound Analysis of Localization Using Signal Strength Difference as Location Fingerprint. In *INFOCOM*, 2010.

[54] S. Wang, R. Inkol, and B. R. Jackson. Relationship Between The Maximum Likelihood Emitter Location Estimators Based on Received Signal Strength (RSS) and Received Signal Strength Difference (RSSD). In *Communications (QBSC), 2012 26th Biennial Symposium on*, pages 64–69, May 2012.

[55] S. Wang and R. Inkol. A near-optimal least squares solution to received signal strength difference based geolocation. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 2600–2603, May 2011.

[56] Bo-Chieh Liu, K. H. Lin, and Jieh-Chian Wu. Analysis of hyperbolic and circular positioning algorithms using stationary signal-strength-difference measurements in wireless communications. *IEEE Transactions on Vehicular Technology*, 55(2):499–509, March 2006.

[57] H. Lohrasbipeydeh, T. A. Gulliver, and H. Amindavar. Blind Received Signal Strength Difference Based Source Localization With System Parameter Errors. *IEEE Transactions on Signal Processing*, 62(17):4516–4531, Sept 2014.

[58] X. Li. RSS-Based Location Estimation with Unknown Pathloss Model. *IEEE Transactions on Wireless Communications*, 5(12):3626–3633, December 2006.

[59] Y. Xu, J. Zhou, and P. Zhang. RSS-Based Source Localization When Path-Loss Model

Parameters are Unknown. *IEEE Communications Letters*, 18(6):1055–1058, June 2014.

[60] Y. T., B. H., R., and F. Estimation of emitter power, location, and path loss exponent. In *Electrical Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*, pages 1–5, April 2012.

[61] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, Oct 2000.

[62] L. Doherty, K. S. J. pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655–1663 vol.3, 2001.

[63] D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, volume 5, pages 2926–2931 vol.5, 2001.

[64] N. Sundaram and P. Ramanathan. Connectivity based location estimation scheme for wireless ad hoc networks. In *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, volume 1, pages 143–147 vol.1, Nov 2002.

[65] Arthur M. Breipohl K. Sam Shanmugan. *Random Signals: Detection, Estimation and Data Analysis*. Wil, 1988.

[66] N. Patwari and P. Agrawal. Effects of Correlated Shadowing: Connectivity, Localization, and RF Tomography. In *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, pages 82–93, April 2008.

[67] K. Yang, J. An, X. Bu, and G. Sun. Constrained Total Least-Squares Location Algorithm Using Time-Difference-of-Arrival measurements. *IEEE Transactions on Vehicular Technology*, 59(3):1558–1562, March 2010.

[68] David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer Publishing Company, Incorporated, 2015.

[69] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

[70] K. C. Ho and M. Sun. An Accurate Algebraic Closed-Form Solution for Energy-Based

Source localization. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(8):2542–2550, Nov 2007.

[71] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, March 1996.

[72] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[73] H. A. Nguyen, H. Guo, and K. S. Low. Real-Time Estimation of Sensor Node's Position Using Particle Swarm Optimization with Log-Barrier Constraint. *IEEE Transactions on Instrumentation and Measurement*, 60(11):3619–3628, Nov 2011.

[74] A. O. Hero and D. Blatt. Sensor network source localization via projection onto convex sets (POCS). In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 3, pages iii/689–iii/692 Vol. 3, March 2005.

[75] Q. Shi and C. He. A new incremental optimization algorithm for ML-based source localization in sensor networks. *IEEE Signal Processing Letters*, 15:45–48, 2008.

[76] M.G. Rabbat and R.D. Nowak. Decentralized source localization and tracking [wireless sensor networks]. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 3, pages iii–921–4 vol.3, May 2004.

[77] Y. Liu, Y. H. Hu, and Q. Pan. Distributed, robust acoustic source localization in a wireless sensor network. *IEEE Transactions on Signal Processing*, 60(8):4350–4359, Aug 2012.

[78] T. Panigrahi, G. Panda, and B. Mulgrew. Distributed bearing estimation technique using diffusion particle swarm optimisation algorithm. *Wireless Sensor Systems, IET*, 2(4):385–393, December 2012.

[79] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, Nov 1995.

[80] R. Hassan, B. Cohanim, O. de Weck, and G. Venter. A comparison of particle swarm optimization and the genetic algorithm. *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Structures, Structural Dynamics, and*

*Materials and Co-located Conferences*, 2005.

[81] M. G. Rabbat, R. D. Nowak, and J. A. Bucklew. Generalized consensus computation in networked systems with erasure links. In *Signal Processing Advances in Wireless Communications, 2005 IEEE 6th Workshop on*, pages 1088–1092, June 2005.

[82] DP Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Acedemic Press, 1982.

[83] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533, Sept 2004.

[84] Panda G. Mulgrew B. Majhi B. Panigrahi, T. Maximum Lilkelihood Source Localization in Wireless Sensor Network Using Particle Swarm Optimization. *Proc. Int. Conf. on electronics Systems (ICES)*, pages 111–115, 2011.

[85] D. Bratton and J. Kennedy. Defining a Standard for Particle Swarm Optimization. In *2007 IEEE Swarm Intelligence Symposium*, pages 120–127, April 2007.

[86] T. Hendtlass. WoSP: a multi-optima particle swarm algorithm. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 727–734 Vol.1, Sept 2005.

[87] Ruixin Niu and P.K. Varshney. Target Location Estimation in Sensor Networks With Quantized Data. *Signal Processing, IEEE Transactions on*, 54(12):4519–4528, Dec 2006.

[88] O. Ozdemir, R. Niu, and P.K. Varshney. Channel Aware Target Localization With Quantized Data in Wireless Sensor Networks. *Signal Processing, IEEE Transactions on*, 57(3):1190–1202, March 2009.

[89] Shu-Jun Liu Zhenxing Luo, Paul S. Min. Target localization in wireless sensor networks for industrial control with selected sensors. *International Journal of Distributed Sensor Networks*, 2013.

[90] A. Nedic, A. Olshevsky, A. Ozdaglar, and J.N. Tsitsiklis. Distributed subgradient methods and quantization effects. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4177–4184, Dec 2008.

[91] B. Bjar, P. Belanovic, and S. Zazo. Distributed Gauss-Newton method for localization in Ad-hoc networks. In *2010 Conference Record of the Forty Fourth Asilomar Conference on*

*Signals, Systems and Computers*, pages 1452–1454, Nov 2010.

[92] B. Johansson, C. M. Carretti, and M. Johansson. On distributed optimization using peer-to-peer communications in wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, pages 497–505, June 2008.

[93] Ozdaglar A. Nedic, A. *Convex optimization in signal processing and communications*, chapter Cooperative distributed multi-agent optimization, pages 240–386. Cambridge University Press, 2008.

[94] D. Thanou, E. Kokiopoulou, Y. Pu, and P. Frossard. Distributed Average Consensus With Quantization Refinement. *Signal Processing, IEEE Transactions on*, 61(1):194–205, Jan 2013.

[95] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, June 2006.

[96] Paolo Frasca, Ruggero Carli, Fabio Fagnani, and Ro Zampieri. Average consensus on networks with quantized communication. *Intern. Journ. on Non-linear and Robust Control*, 19:1787–1816, 2008.

[97] Lin Xiao, Stephen Boyd, and Sanjay Lall. Distributed Average Consensus with Time-Varying Metropolis Weights, 2006.

[98] Russell Merris. Laplacian matrices of graphs: a survey. *Linear Algebra and its Applications*, 197(Supplement C):143 – 176, 1994.

[99] M.A. Matin and M.M. Islam. Overview of wireless sensor network. In Mohammad A. Matin, editor, *Wireless Sensor Networks - Technology and Protocols*, chapter 01. InTech, Rijeka, 2012.

[100] Kazem Sohraby, Daniel Minoli, and Taieb Znati. *Wireless Sensor Networks: Technology, Protocols, and Applications*. Wiley-Interscience, 2007.

[101] Amitabha Ghosh and Sajal K. Das. Coverage and connectivity issues in wireless sensor networks: A survey. *Pervasive and Mobile Computing*, 4(3):303 – 334, 2008.

[102] Jaehyun Park Stephen Boyd, Steven Diamond. Convex optimization overview. Technical report, EE & CS Departments, Stanford University, 2016.

[103] P. A. Forero, A. Cano, and G. B. Giannakis. Consensus-based distributed expectation-maximization algorithm for density estimation and classification using wireless sensor networks. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 1989–1992, March 2008.

[104] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *Decision and Control, 2007 46th IEEE Conference on*, pages 5492–5498, Dec 2007.

[105] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

[106] Federica Garin and Luca Schenato. *A Survey on Distributed Estimation and Control Applications Using Linear Consensus Algorithms*, pages 75–107. Springer London, London, 2010.

[107] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65 – 78, 2004.

[108] A. Nedic and A. Ozdaglar. Distributed Subgradient Methods for Multi-Agent Optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, Jan 2009.

[109] D. Jakovetic, J. Xavier, and J. M. F. Moura. Fast Distributed Gradient Methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, May 2014.

[110] Yiguang Honga Peng Yi. Distributed gradient algorithm for constrained optimization with application to load sharing in power systems. *Systems & Control Letters*, 83:45C52, 2015.

[111] Yuhui Shi and R. Eberhart. A modified particle swarm optimizer. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 69–73, May 1998.

[112] Yuhui Shi and R.C. Eberhart. Empirical study of particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3, page 1950 Vol. 3, 1999.